AD-A099 108 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 20/4
A USER'S MANUAL FOR THE DTNSRDC MOMENTUM INTEGRAL BOUNDARY-LAYE--ETC(U)
DEC 80 T J LANGAN, C VON KERCZEK
UNCLASSIFIED DTNSRDC/SPD-0972-01 NL

1 OF 2
AD
A099108

LEVEL II (12)

AD A099108

DTIC FILE COPY

A USER'S MANUAL FOR THE DTNSRDC MOMENTUM INTEGRAL BOUNDARY-LAYER CODE FOR SHIP HULLS DTNSRDC/SPD-0972-01

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Maryland 20084

A USER'S MANUAL FOR THE DTNSRDC MOMENTUM INTEGRAL BOUNDARY-LAYER CODE FOR SHIP HULLS

by

Thomas J. Langan

Christian von Kerczek

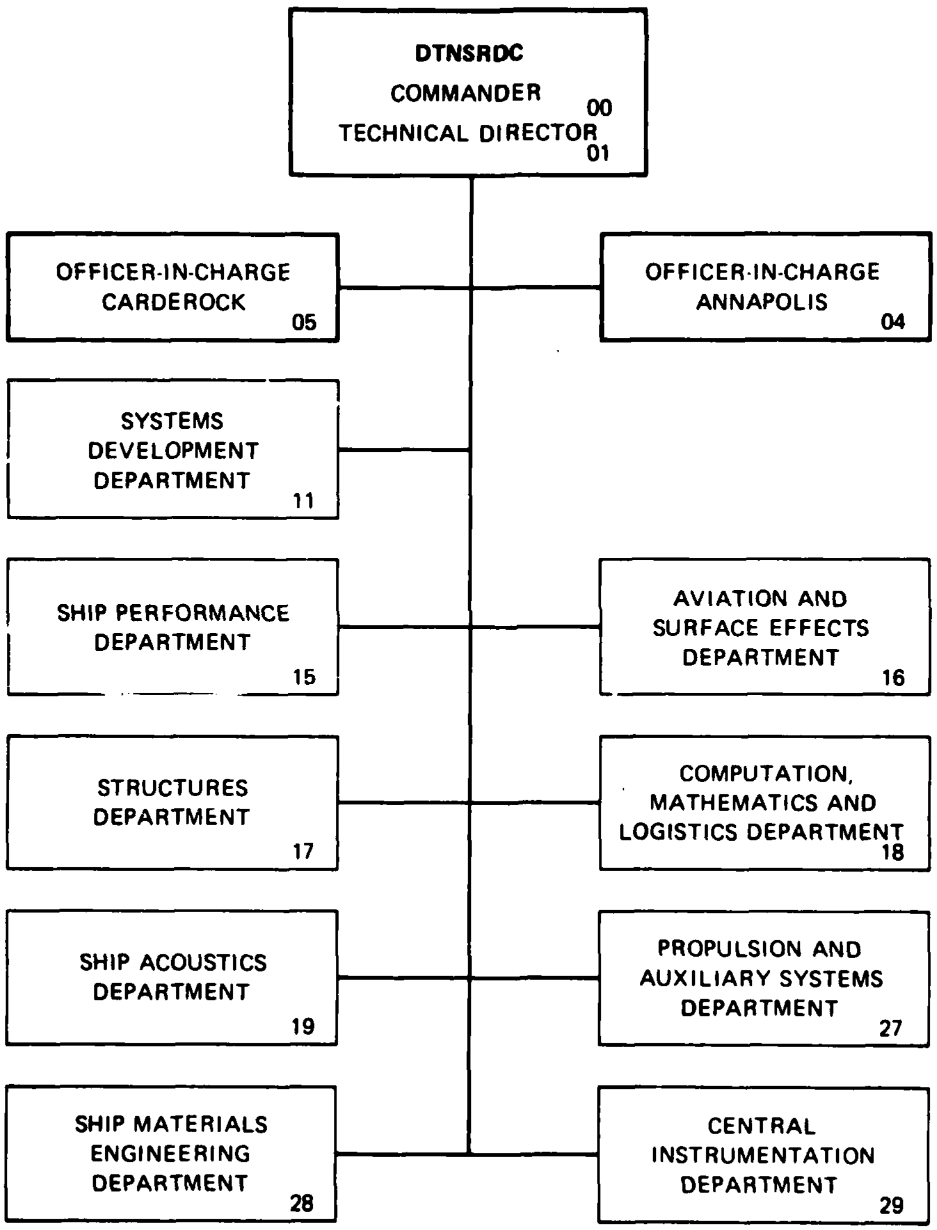
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

SHIP PERFORMANCE DEPARTMENT

DEPARTMENTAL REPORT

December 1980 DTNSRDC/SPD-0972-01

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 DTNSRDC/SPD-0972-01	2. GOVT ACCESSION NO. 9	3. RECIPIENT'S CATALOG NUMBER Final rept.
4. TITLE (and Subtitle) 6 A User's Manual for the DTNSRDC Momentum Integral Boundary-Layer Code for Ship Hulls.	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Thomas J. Langan, Christian von Kerczek	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship Research and Development Center, Bethesda, Maryland 20084	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS See Reverse Side
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command (SEA-035) Washington, D.C. 20362	12. REPORT DATE 11 Dec 1980
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 160	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved For Public Release: Distribution Unlimited

16 SF43301 17 SR0230101

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Boundary Layers
Momentum Integral
Turbulent
Three-Dimensional

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report provides the instructions for using a suite of computer codes to calculate approximately the three-dimensional boundary layer on ship hulls. There are two sets of codes for computing boundary layers. One is for computing a first-order approximation and is based on a slender-body potential and the small crossflow assumption. The second set of codes uses a general three-dimensional calculation method for the potential flow and solves the three-dimensional momentum integral boundary-layer equations

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

389694

Block #20

without making restrictive assumptions on the magnitude of the crossflow. An example problem is presented for both methods.

Block #10

Project Element	61153N 23
Project Number	SR 02301
Task Area	SR 023-01-01
Work Unit	1552-070

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This report provides the instructions for using a suite of computer codes to calculate approximately the three-dimensional boundary layer on ship hulls. There are two sets of codes for computing boundary layers. One is for computing a first-order approximation and is based on a slenderbody potential and the small crossflow assumption. The second set of codes uses a general three-dimensional calculation method for the potential flow and solves the three-dimensional momentum integral boundary-layer equations without making restrictive assumptions on the magnitude of the crossflow. An example problem is presented for both methods.

ADMINISTRATIVE INFORMATION

The work supported herein was supported by the General Hydromechanics Research Program under Task Area SR 023-01-01 and Work Unit 1552-070.

I. INTRODUCTION

This manual provides the instructions for using a suite of computer codes to calculate approximately the three-dimensional turbulent boundary layer on ship hulls. The development of the methods programmed into this suite of boundary-layer codes, from here on designated BLC for the entire system of programs, is given in five references (1-5).

The package BLC has two sets of codes. The first set of codes is for computing a first-order approximation to the three-dimensional boundary layer on fairly fine ships such as a Series 60, Block 60 hull. These codes use slender-body-theory potential flow[1], at zero Froude number, to obtain the input pressure distribution. Secondly, the three-dimensional momentum integral equations with the assumption of small crossflow in the boundary layer are solved in streamline coordinates[2]. This is another reason restricting these codes to slender ships. This set of codes is very easy to use and inexpensive to run on a computer so that for a first estimate of the ship boundary layer, it may be useful to employ them. Run time is less than two minutes for this code, whereas, twenty to thirty minutes is required by the second set of codes.

The second set of codes in BLC is more general in that the slender-body-theory

potential flow calculation method is replaced by a general three-dimensional calculation method. Furthermore, the three-dimensional momentum integral boundary-layer equations are solved in a general orthogonal surface coordinate system without making restrictive assumptions on the magnitude of the crossflow.

We shall briefly describe below the three main segments of BLC which make up the second, more general, boundary-layer calculation method. The geometry program represents the ship hull by a surface equation and is common to both the first and second set of programs in BLC.

The code for representing the ship hull by a surface equation from which all the necessary geometrical data are computed is based on the theory published by von Kerczek and Tuck[3]. The potential flow data is computed by a code based on the surface doublet distribution that was developed by Chang and Pien[4], and the three-dimensional turbulent boundary layer is computed by the momentum integral method described by von Kerczek and Langan[5]. We refer the reader to references (3) - (5) for further details. We emphasize at the outset that the BLC provides only approximate results that are valid for (i) ships moving straight ahead at zero Froude number, i.e., the free surface is assumed to be plane, no waves are taken into account; (ii) the B.L. flow is unseparated and in fact very thin compared with the local radius of curvature in any direction on the body, so that the stern region is not properly modeled; and (iii) the boundary layer is fully turbulent, i.e., the laminar and transitional parts of the boundary layer near the bow are not modelled.

The underwater part of the ship hull is represented by a surface equation that consists of the conformal mapping of a unit circle onto the ship section and the polynominal interpolation of these sectionwise mappings along the length of the ship (see reference (1)). This type of surface representation can be used for most conventional ship hull shapes but it excludes two important geometrical features often found on ships. The first exclusion is a forward protruding bulb. The second exclusion is the rear portion of cruiser-type sterns. These exclusions of geometric forms are acceptable since the boundary-layer calculation method is not applicable in these regions of the ship.

The potential flow code[4] is general and will handle any nonlifting fully-submerged body with a vertical plane of symmetry. The restriction to submerged bodies means, for the present situation, a double ship hull in which free surface effects are neglected. The code is based on the calculation of the doublet strength density

on the surface of the ship that will result in potential flow tangency on the surface. The method consists of distributing constant strength doublets on an assemblage of quadrilateral surface elements that approximate the actual surface of the ship and then solving the resulting algebraic equations that approximate the integral equation.

The numerical solution of the boundary-layer equations is carried out by a three-dimensional momentum integral-entrainment method[5]. In this method, the boundary-layer crossflow can be large but the accuracy of the computer results is limited by the very rough approximations that are made for the crossflow profile shape, the rate of entrainment coefficient and the relationships assumed for the crossflow parameters as functions of the streamwise flow parameters.

It is possible to give only rough estimates of the level of accuracy of the computed results of this program becasue the accuracy varies not only from one hull to another, but also on various parts of one hull. However, we have shown that on one model hull, the SSPA-720 Model, block coefficient 0.675, for which detailed experimental boundary layer data[6] were available, the computed values of streamwise momentum and displacement thicknesses and surface shear stresses that were computed by the second system of codes in BLC were mostly within 10 percent of the experimental data[5]. Larger discrepancies between the computed and experimental results occurred only in localized regions of the hull. For example, very close to the stern (within the last 5 percent of the ship length), the experimentally determined boundary layer becomes very thick and the surface shear stress approaches the value of zero; the computed values of shear stress tend to overpredict the experimental values by as much as 100 percent in this region. Furthermore, there is a qualitative discrepancy between the computed and experimental values of displacement and momentum thicknesses and surface shear stresses along a streamline that lies very close to the keel of the hull. However, in general, the computed results from the second set of boundary-layer codes in BLC will be qualitatively correct and quantitatively will give values of about 90 percent of the experimental values for the streamwise displacement and momentum thicknesses and surface shear stresses on about 80 percent of the hull length, exclusive of the ends.

The crossflow angle of β that measures the difference in direction of the surface shear stress vector and the local direction of the inviscid surface streamline at the surface is not well-predicted by the methods of BLC. Over most of the hull

surface, where β is fairly small, the predictions of β from the BLC are qualitatively correct. However, near the stern of the model tested by Larsson[6], where the experimental values of β become a little more substantial, on the order of 10-20 deg, the predicted values of BLC are substantially different, sometimes even having the wrong sign.

The numerical methods that are used in BLC are well-established and have been tested for accuracy as a function of step or panel sizes. The recommended step and panel sizes given in Chapter 3 in conjunction with the descriptions of the input schemes have been ascertained to yield at least a 2-3 percent level of numerical accuracy. Further reductions of step and panel sizes do reduce this error accordingly.

We need to mention here that empirical boundary-layer data are needed to start the calculations at some station downstream of the bow stern. The effects of inaccuracies in this starting data on the subsequent boundary-layer development are not known, but it seems that, if the starting station is located in a favorable pressure gradient region, then the downstream development of the boundary layer is not too sensitive to the starting conditions. Some experimentation will be necessary to ascertain this for individual cases.

The rest of this manual is organized as follows. Chapter 2 gives a complete description of each element of BLC and its relationship to the rest of the programs in BLC. In Chapter 3 we present a detailed discussion of the operation of the programs in conjunction with an example. The example is the SSPA Model 720, for which extensive boundary layer data exist[6]. A listing of the program is given in the appendix.

2. PROGRAMS

2.1 MAP22

The program MAP22 fits an analytic function to a set of hull offsets which are specified at K stations. It fits each of the K cross sections with an N-parameter map of the unit circle given by the equation[2]:

$$Z = \sum_{n=1}^{N} a_n \zeta^{3-2n} \tag{1}$$

Figure 1 shows the relationship between the complex Z and ζ-planes and also shows a typical map; Figure 2 shows the hull coordinate system. In the program the mapping is split into its real and imaginary parts; so at the I-th station it takes the form

$$
\begin{aligned}
\bar{X} &= \sum_{J=1}^{N} AN(I,J) \cos (3-2J)\theta \\
Y &= \sum_{J=1}^{N} AN(I,J) \sin (3-2J)\theta
\end{aligned} \qquad (2)
$$

Once the AN's are computed, the program fits a polynominal in S to the parameters for J=1, 2, ...,N. It computes the coefficients BN(L,J) of these polynomials by solving the linear system

$$
AN(I,J) = \sum_{L=1}^{KPK} BN(L,J) * S(I) **(L-1) \qquad (3)
$$

Each polynomial has KPK coefficients

The hull is represented by the system of equations

$$
\begin{aligned}
X (S,\theta) &= \sum_{I=1}^{KPK} \sum_{J=1}^{N} BN(I,J) * S^{(I-1)} \cos(3-2J)\theta \\
Y (S,\theta) &= \sum_{I=1}^{KPK} \sum_{J=1}^{N} BN(I,J) * S^{I-1} \sin (3-2J)\theta
\end{aligned} \qquad (4)
$$

This representation is used throughout the suite of programs.

The number of offsets given at each of the K stations need not be the same; for the moment let P offsets be given at station I. There are then P points of the unit circle in the fourth quadrant that will be mapped onto the P points of the I contour corresponding to these offsets. Since these points can be arbitrarily chosen, there are P values of θ; namely θ_p (p=1,..., P), to be determined in addition to the N parameters AN (I,J). There are then 2P equations to be solved for

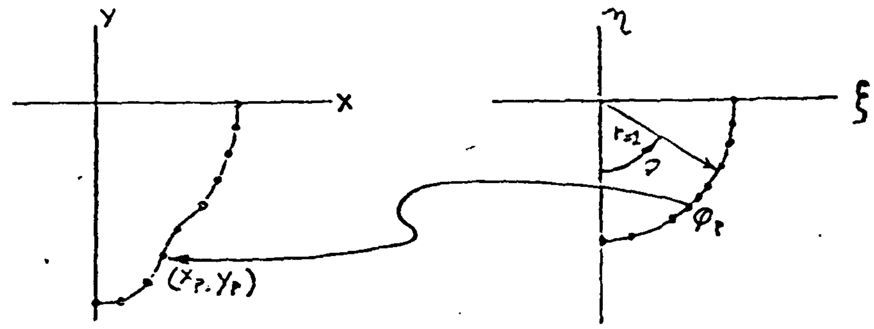


Figure 1 - Mapping of Unic Circle onto a Section

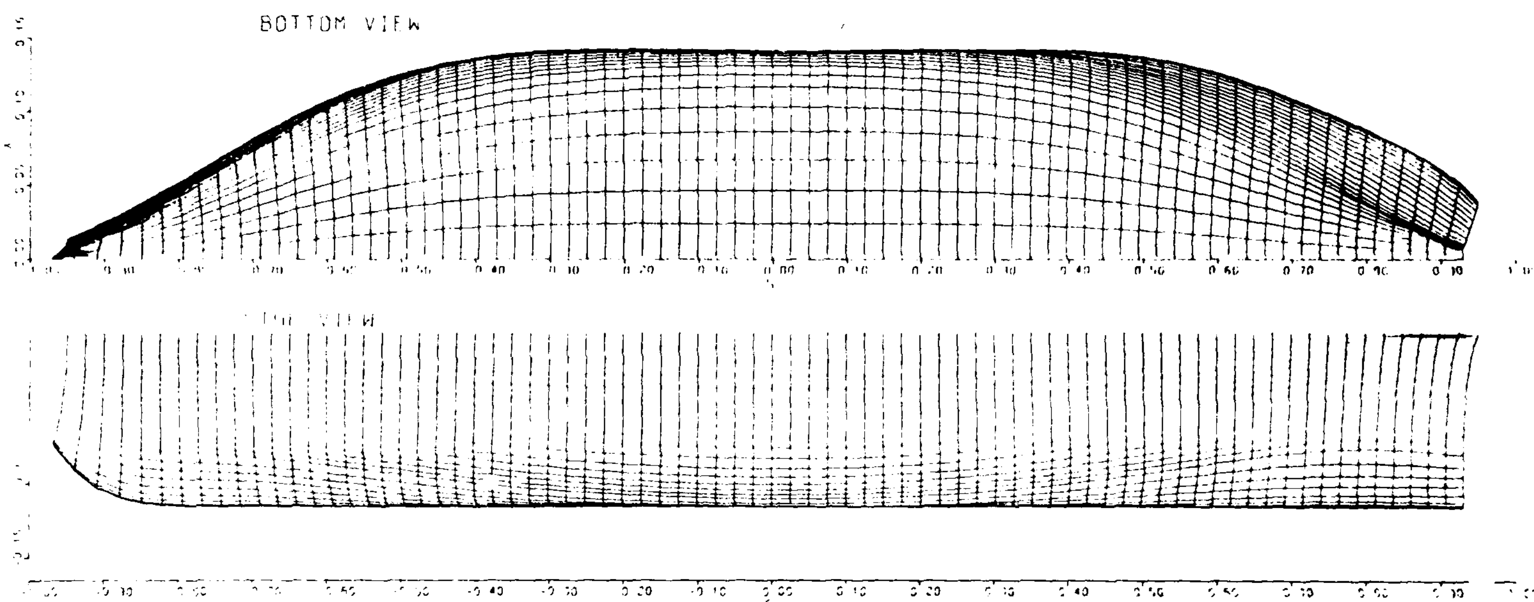


Figure 2 - The Bottom and Side Views of the (ϕ,θ) Coordinate Net on the SSPA 720 Model

the N values AN(I,J) (J=1, ..., N) and the P values of θ_p ; namely ,

$$XMS(I,p) = \sum_{J=1}^{N} AN(I,J) \cos (3-2J)\theta_p$$

$$YMS(I,p) = \sum_{J=1}^{N} AN(I,J) \sin (3-2J)\theta_p \qquad (5)$$

where XMS and YMS are respectively the (x,y) offset of the P-th point. If N=P, unacceptable wriggles occur in the approximating cross section between the specified points; instead, N is taken to be of the order of 1/2 to 2/3 P and Equation (5) is solved in a least squares sense. Specifically, the AN's and θ_p are chosen to minimize the error

$$E = \sum_{p=1}^{P} (E_p'^{\,2} + E_p''^{\,2}) \qquad (6)$$

where

$$E_p' = XMS(I,p) - \sum_{J=1}^{N} AN(I,J) \cos (3-2J)\theta_p$$

$$E_p'' = YMS(I,p) - \sum_{J=1}^{N} AN(I,J) \sin (3-2J)\theta_p$$

Not only does this approach result in a smooth form; it also fits the data points in a manner that minimizes the squared distance between the data points and the generated form. The relations

$$0 = \frac{\partial E}{\partial AN(I,J)} = L\,(AN;\theta_p)$$

$$0 = \frac{\partial E}{\partial \theta_p} = N(\theta_p;AN) \qquad (7)$$

are necessary conditions for E to have a minimum, where L and N are given by von Kerczek and Tuck[2]. The system L =0 is linear in AN and can be solved provided θ_p is given. The system N =0 is totally nonlinear in both variables; however, it

can be used as a system of equations for θ_p provided the AN are assumed.

It is not practical to compute the best N-term fit directly, for this approach allows too much freedom in the AN's and results in a relatively bad fit. Instead, the best 4-term, 5-term, ..., N-term fit is determined successively starting with the 3-term Lewis form. Moreover, the values of AN (I,1) and AN (I,2) are further restricted to conform more closely with the Lewis forms. The best (J-1) term map serves as the initial guess to the j-th term map, which is solved iteratively. The first step of this iteration is to substitute the present values of the θ_p together with AN(I,1) and AN(I,2) into the system L =0. This system is solved by linear methods for the j-2 terms AN(I,3), ..., AN(I,j). Next the program computes AN(I,1) and AN(I,2) using the equations

$$A(I,1) = 1/2 \{ B(I) + H(I) - \sum_{J=3}^{j} [1-(-1)^J] AN (I,J) \}$$

$$A(I,2) = 1/2 \{ B(I) - H(I) - \sum_{J=3}^{j} [1 + (-1)^J] AN (I,J) \}$$

where B(I) is the beam at station I and H(I) is the draft. System N=0 is then solved iteratively for the θ_p. If the resultant value for E has not changed from the previous value by more than a given tolerance TOL, the resulting AN's and θ_p are taken as the value for the best j-term fit; otherwise , the first step of the iteration is entered again.

MAP22 uses a least squares method to solve Equation (3) for the KPK coefficients BN(L,J), ..., BN(KPK,J). It has the facility to constrain the polynomial at any of the K stations at which the offsets are specified. The BN can be determined such that at station I the approximating polynomial has the same value as AN(I,J), zero slope, both zero slope and curvature, or a combination. An additional feature of the program is the provision made to partition the hull surface into a forebody, a parallel middlebody, and an afterbody. The forebody and afterbody are each constrained at their junction with the middlebody to have zero slope and curvature in the longitudinal direction and to have the same shape as the middlebody cross section.

2.2 TBL18

This program is based on the method of von Kerczek [2] for computing a first-order approximation to the three-dimensional boundary layer on a fairly fine ship hull. The program uses the analytic surface representation, Equation (4), for the hull geometry and applies the slender-body potential derived by Tuck and von Kerczek to calculate the inviscid flow. It applies the Cumpsty-Head-Smith[7,8] three-dimensional turbulent boundary-layer calculation method to compute the entrainment, streamwise momentum, and skin friction along given streamlines. It is assumed that the crossflow is small in these calculations.

The Tuck and von Kerczek formula for the complex potential $f(s,\zeta)$ is given by

$$f(s,\zeta) = CN(1) \log AN(1)\, \zeta + BO - \sum_{I=1}^{N-1} \frac{CN\,(I+1)}{2I} \tag{8}$$

This is the potential for the inviscid flow around the analytic surface

$$AN(J) = \sum_{L=1}^{KPK} BN(L,J)\; S^{L-1} \tag{9}$$

which agrees with (4) for $S = S(I)$. The functions CN are given by

$$CN(1) = \sum_{J=1}^{N} (3-2J)\; AN(J) * ANP(J) \tag{10}$$

and

$$CN(J) = \sum_{L=1}^{N} (3-2L)\; AN(L)\; [ANP(L+J) + ANP(L-J)] \tag{11}$$

where ANP is the derivative of AN with respect to S ($ANP(J) \equiv 0$ for $J < 1$ or $J > N$). Finally the function BO of S is given as

$$BO = -\tfrac{1}{2}\, CN(1) * LOG\,[\, 4(1-S^2)\,] + \tfrac{1}{2} \int_{-1}^{1} \frac{C_o(s) - C_o(\xi)}{|S - \xi|} \tag{12}$$

where $C_o(S) = CN(1)$ evaluated at S. The integral is evaluated as a sum of terms containing the integrals

$$\int_{-1}^{1} \frac{S^j - \xi^j}{|S - \xi|}\, d\xi = 2\left(\sum_{k=1}^{j} \frac{1}{k} \right) S^j - \sum_{k=1}^{j} \frac{1 + (-1)^k}{k}\, S^{j-k} \tag{13}$$

for J = 1, ..., N.

The integration of the boundary-layer equations begin at station S(1), where empirical boundary-layer data are given as initial data for the differential equations. These boundary layer equations are

$$\frac{d\theta_{11}}{d\alpha} + \theta_{11}(2 + H)\frac{1}{U_s}\frac{dU_s}{d\alpha} - \theta_{11}K_1 = C_{f_1}$$

$$\frac{d\theta_{21}}{d\alpha} + 2\theta_{21}\left(\frac{1}{U_s}\frac{dU_s}{d\alpha} - K_1\right) + \theta_{11}K_2(1 + H) = C_{f_2} \qquad (14)$$

$$\frac{d(\theta_{11}G)}{d\alpha} + \theta_{11}G\left(\frac{1}{U_s}\frac{dU_s}{d\alpha} - K_1\right) = F(G)$$

where G is given in terms of the boundary-layer shape parameter H by

$$G = 1.535\ (H - 0.7)^{-2.715} + 3.3 \qquad (15)$$

and

$$F(G) = 0.0306\ (G-3.0)^{-01653} \qquad (16)$$

$d\alpha$ is the differential element of arc along a streamline; K_1 and K_2 are geodesic curvatures; and U_s is the inviscid fluid velocity at the wall.

$$\theta_{11} = \int_o^\delta \left(1 - \frac{u}{U_o}\right)\frac{u}{U_o}\, d\lambda \qquad (17)$$

$$\theta_{21} = -\int_o^\delta \frac{uv}{U_o^2}\, d\lambda \qquad (18)$$

where u is the streamwise velocity in the boundary layer, and v is the cross-flow velocity. The friction coefficients C_{f_1} and C_{f_2} are in the streamwise and cross-flow directions respectively.

An equal step size Euler method is used in the numerical solution of Equation (14).

2.3 TBL3D

This program is a modification of the potential portion of TBL18 and is based on the same hull representation, Equation (4). Its primary purpose is the computation of the surface coordinates, distance metrics, and geodesic curvatures used in the three-dimensional momentum integral equations.

The intended application of BLC calls for the use of the inviscid velocities obtained from using a doublet distribution over the hull. Additionally, BLC has been provided with a quick way to estimate the soultion to the three-dimensional momentum integral equation. This solution uses the Tuck and von Kerczek slender body potential to compute the inviscid velocities at the grid points. The potential portion of these computations is carried out in TBL3D.

If r (S,θ) is the vector from the origin of the (X,Y,S) coordinate system, Figure (2)

$$\underline{r}\,(S,\ \theta) = X(S,\theta)\underline{i} + Y(S,\theta)\underline{j} + S\,\underline{k} \tag{19}$$

($\underline{i}$, $\underline{j}$, $\underline{k}$) are the unit tangent vectors to the (X, Y, S) coordinate systems respectively. The surface coordinate lines θ = constant, run along the length of the hull surface; the orthogonal trajectories to these coordinate lines form the coordinate lines ϕ = constant. These latter coordinate lines are nearly parallel to the hull cross-sections. The arc length increments along the ϕ and θ coordinates are given by von Kerczek and Langan[5] as

$$d\ell\phi = (d\underline{r}\ .\ d\underline{r})^{\frac{1}{2}}_{\theta\,=\,\text{constant}} = H(S,\theta)\ dS \tag{20}$$

and

$$d\ell\theta = (d\underline{r}\ .\ d\underline{r})^{\frac{1}{2}}_{\phi\,=\,\text{constant}} = G(S,\theta)d\theta \tag{21}$$

respectively, where

$$H(S,\theta) = \left[(\frac{\partial X}{\partial S})^2 + (\frac{\partial Y}{\partial S})^2 + 1\right]^{\frac{1}{2}} \tag{22}$$

$$G(S,\theta) = \left[(\frac{\partial X}{\partial \theta} + \frac{\partial X}{\partial S} + F)^2 + (\frac{\partial Y}{\partial \theta} + \frac{\partial Y}{\partial S} + F)^2 + F^2\right]^{\frac{1}{2}} \tag{23}$$

and

$$F = \left.\frac{dS}{\partial\theta}\right|_{\phi} = \frac{-\left(\frac{\partial X}{\partial \theta}\ \frac{\partial X}{\partial S} + \frac{\partial Y}{\partial \theta}\ \frac{\partial Y}{\partial S}\right)}{\left[1 + (\frac{\partial X}{\partial \theta})^2 + (\frac{\partial Y}{\partial \theta})^2\right]} \tag{24}$$

The program uses the unit orthogonal vectors $\underline{e}_\phi$, $\underline{e}_n$, $\underline{e}_\theta$, to simplify the computation of the geodesic curvatures K_ϕ and K_θ. Since the ϕ-coordinates are generated by holding θ constant in Equation 19 for $\underline{r}$, the unit vector in the ϕ-direction is given by

$$\underline{e}_\phi' = (EP1,\ EP2,\ EP3) = H^{-1} \left(\frac{\partial X}{\partial S} ,\ \frac{\partial Y}{\partial S} ,\ 1 \right) \tag{25}$$

The unit vector normal to the surface is determined from the cross product $\frac{\partial \underline{r}}{\partial \theta} \times \frac{\partial \underline{r}}{\partial S}$ which is normal to the surface of the hull; hence,

$$\begin{aligned} \underline{e}_n &= (EN1,\ EN2,\ EN3) \\ &= D^{-1} \left(\frac{\partial Y}{\partial \theta} ,\ - \frac{\partial X}{\partial \theta},\ \frac{\partial X}{\partial \theta} \frac{\partial Y}{\partial S} - \frac{\partial X}{\partial S} \frac{\partial Y}{\partial \theta} \right) \end{aligned} \tag{26}$$

where

$$D = \left[\left(\frac{\partial Y}{\partial \theta} \right)^2 + \left(\frac{\partial X}{\partial \theta} \right)^2 + \left(\frac{\partial X}{\partial \theta} \frac{\partial Y}{\partial S} - \frac{\partial X}{\partial S} \frac{\partial Y}{\partial \theta} \right)^2 \right]^{\frac{1}{2}} \tag{27}$$

From the definition of orthonormal

$$\underline{e}_\theta = \underline{e}_n \times \underline{e}_\phi ; \quad | \underline{e}_\theta | = 1$$

that is

$$\begin{aligned} \underline{e}_\theta &= (ET1,\ ET2,\ ET3) \\ &= (EN2 * EP3 - EN3 * EP2,\ EN3 * EP1 \\ &\quad - EN1 \times EP3,\ EN1 * EP2 - EN2 * EP1) \end{aligned} \tag{28}$$

The geodesic curvatures are calculated using

$$K_\phi = \underline{e}_\phi \cdot \frac{d\underline{e}_\theta}{d\ell_\theta} = - \frac{1}{h_\theta h_\phi} \frac{\partial h_\theta}{\partial \phi} \tag{29}$$

and

$$K_\theta = \underline{e}_\theta \cdot \frac{d\underline{e}_\phi}{d\ell\phi} = - \frac{1}{h_\phi h_\theta} \frac{\partial h_\phi}{\partial h_\theta} \tag{30}$$

2.4 CBL

The program CBL lays out the panels for the doublet distribution on the analytic hull. It merely repeatedly applies Equation (9) to obtain the coordinates for the panel corners and formats this data for the potential flow program DOUBLD.

If one is simply computing the potential flow around a double hull model, one needs only to use MAP22, CBL, and DOUBLD.

2.5 DOUBDD

The program DOUBDD computes the potential and velocities for inviscid flow around a double hull model. Pien and Chang[4] developed the original form of this program; with their permission we have documented their program here to publicize it and to have a complete discussion of the program system BLC. The program solves the Neumann problem for flow around a body by distributing a layer of doublets on the surface of the body, in our case the double hull, and solving numerically the resulting integral equation by a panel method. The magnitude of the doublet distribution on the panels, SIG(I), is obtained directly as the soultion of the linear system

$$\sum_{J=1}^{NP} \mathrm{VPP(I,J)} * \mathrm{SIG(J)} = -4\Pi\, \overline{S}\,(I) \tag{31}$$

for I=1, ..., NP, which approximates the integral equation

$$4\Pi\, S(P) = \iint_{\substack{\text{Double}\\ \text{Hull}}} \sigma(Q)\, G(P,Q)\, ds_Q$$

SIG(J) is the strength of the doublets distributed on the J-th panel, VPP(I,J) is the influence of this doublet distribution on the I-th panel, and $\overline{S}(I)$ is an average station location for the I-th panel. In the integral equation G(P,Q) is the three-dimensional distribution 1/r, where r is the distance between the points P and Q on the hull surface, and ds_Q is the increment of surface arca used in the integration.

The doublet distribution is the perturbation potential due to the inviscid flow around the double hull.

In the actual mechanics of the solution DOUBDD replaces the curva-linear panels, that are constructed in CBL, by plane panels. These planes are chosen to minimize

the distance between the corners of the plane panels and those of the curvilinear ones. If the longitudinal length of a plane panel exceeds its traverse width by a multiple of two or more the panel length is divided into subpanels which satisfy the criterion that their length is less than twice their width.

The off-diagonal elements of the influence matrix VPP are computed in a straight forward manner. This matrix can be ill conditioned, and a straight forward computation of the diagonal elements V(I, I) can lead to serious error. A different approach is taken in the computation of the diagonal elements to avoid the problem. Since the equation

$$\sum_{J=1}^{NP} VPP(I,J) = -4\Pi \qquad (32)$$

corresponds to Greens identity

$$SS \frac{\partial}{\partial n} G(P,\theta)\, dS_Q = -4\Pi$$

one can use

$$VPP(I,I) = -4\Pi - \sum_{\substack{J=1 \\ J\neq I}}^{NP} VPP(I,J) \qquad (33)$$

to compute the diagonal elements. This calculation method does not produce the errors in the solution that results from computing the diagonal elements in a straight forward manner. It is therefore used in DOUBDD.

2.6 TBLIPC

This program organizes the geometric and inviscid flow data for use in the turbulent boundary-layer calculations. These calculations are made on a surface grid (ϕ_i, θ_j) given by

$$\phi_i = \phi_{i-1} + (i-1)\Delta\phi \qquad i = 2, \ldots, NS$$

$$\theta_j = (j-1)\Delta\theta \qquad j = 1, \ldots, NTH$$

The initial value ϕ_1 is the value of ϕ at the station at which the boundary-layer calculations are initiated. Empirical values of the momentum integral θ_{11}, the crossflow angle β, and of the shape parameter H are given at $\phi = \phi_1$ as initial conditions; the calculations proceed station by station downstream from ϕ_1. The bound-

ary-layer equations are first solved for θ_{11}, $t = \tan \beta$ and H at $\phi = \phi_2$; these calculations are then repeated at $\phi = \phi_3, \phi_4, \ldots, \phi_{NS}$. Because the total amount of geometric and inviscid flow data is prohibitively large for storage in main memory and since it is only used station by station, TBLIPC organizes it as a function of θ for each value ϕ_i and stores it on disk for $i = 1, \ldots, NS$.

Typically, DOUBDD computes the inviscid flow potential at 250 to 500 points approximately evenly distributed over the portion of the hull having $0 \geq \theta \geq -\frac{\Pi}{2}$; the boundary layer calculations require the inviscid velocities and their derivatives at 1500 or more points on the grid (ϕ_i, θ_j). TBLIPC uses spline on spline interpolation to determine the potential at the grid points. It uses the derivatives of these spline functions to compute the inviscid velocity tangent to the surface in the ϕ-direction, u_ϕ, and the corresponding velocity in the θ-direction, u_θ.

2.7 TBLSOL

von Kerczek and Langan[5] expressed the momentum integral equations and the entrainment equation in the form

$$A\,(W\,(\phi,\,\theta))\,\frac{\partial W}{\partial \ell_\phi} + B(W(\phi,\theta))\,\frac{\partial W}{\partial \ell_\theta} = C(W)$$

where

$$W = \begin{pmatrix} \theta_{11} \\ t \\ H \end{pmatrix}$$

The program TBLSOL uses an implicit marching method to solve this system of nonlinear equations. In marching from station ϕ_i to ϕ_{i+1} ..., it uses the value of W at (ϕ_i, θ_i) to compute the coefficients A, B, and C at (ϕ_{i+1}, θ_i). The program uses Gaussian reduction to solve the resulting system of linear equations.

The coefficient matrices are described in von Kerczek and Langan[5] together with the theoretical details of the solution method.

3. OPERATING INSTRUCTIONS WITH AN EXAMPLE

The suite of programs BLC has been installed on the Burroughs 7700 at DTNSRDC as an interactive software system. Each individual program was initially designed for batch operation on the CDC 6700 computer system. A version of DOUBDD is pre-

sently available on the Naval Research Laboratory Advanced Scientific Computer; it uses the large array storage and pipeline capabilities of that machine. The individual programs required minimum modification when moved from one machine to the other; they are written in a fairly basic FORTRAN. The operating instructions and program listings presented in this report apply to the interactive software system as installed on the Burroughs 7700.

The Swedish Model SSPA 720 is used in the sample calculations. The offsets used for this ship are given as input to MAP22. The input to each program is given in detail, unless it is generated as output from one of the programs in the system. Since the output from the programs would require too much space and can be easily generated if needed, only partial listings of the output will be given.

3.1 MAP22

This is the first program of the system. Its primary output is the hull geometry representation matrix BN. This matrix forms the basis of all further calculations in the system. The input data is stored in file PFN before running the program. The input data file PFN for the SSPA Model 720 is shown in Table 1. It is read by MAP22 with the following read statements.

```
   READ(5,24) (TITLE(J),J=1,12)
   READ(5,3) IST1,IST2,KP1,KP2
   READ(5,1) K,N,TOL,BL
   READ(5,3) (IFS(I),I=1,K)
   READ(5,3) (JFS(I),I=1,K)
   READ(5,25)(KT(I),I=1,K)
   READ(5,4) (IX(J),IY(J),J=1,M)
    READ(5,2) (S(I),I=1,K)
32  READ(5,2) (AN(I,J),J=1,N)
```

The formats are

```
 1 FORMAT(2I10,2F10.5)
 4 FORMAT(6(1X,2I6))
 2 FORMAT (7F10.5)
 3 FORMAT(14I5)
25 FORMAT(7I10)
24 FORMAT(12A6)
```

TABLE 1 - INPUT TO MAP22

```
#FILE (CHXL)PFN ON DTNSRDC
100         SSPA MODEL 720
200    25     0    11     0
300         25          7   0.0001      1.0
400     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
500     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
600     1
700
800          4          15          17          19          20          23          26
900         29          33          20          21          21          21          21
1000         22          21          19          18          17          16          18
1100         20          21          23           9
1200 *             *        -   500*        - 1000*        - 1636
1300 *&00439-00005*&00407-00272*&00381-00511*&00349-00766*&00331-01002*&00309-01248
1400 *&00287-01503*&00264-01753*&00242-01992*&00217-02236*&00183-02494*&00142-02743
1500 *&00097-02981*&00064-03088*-00001-03091
1600 *&00859-00002*&00823-00271*&00791-00512*&00760-00767*&00736-01007*&00706-01250
1700 *&00686-01508*&00659-01750*&00628-01993*&00585-02235*&00531-02496*&00475-02735
1800 *&00405-02984*&00320-03229*&00223-03476*&00119-03713*-00008-03718
1900 *&01338-00003*&01297-00272*&01263-00514*&01226-00769*&01191-01009*&01168-01253
2000 *&01132-01506*&01094-01754*&01051-01996*&01000-02241*&00932-02493*&00850-02736
2100 *&00747-02987*&00635-03237*&00501-03478*&00348-03718*&00252-03853*&00137-03984
2200 *-00007-03985
2300 *&01837-00004*&01803-00274*&01773-00513*&01732-00771*&01699-01005*&01660-01253
2400 *&01616-01508*&01565-01756*&01506-01996*&01441-02238*&01358-02497*&01270-02735
2500 *&01140-02988*&00990-03233*&00806-03480*&00583-03720*&00427-03873*&00311-03972
2600 *&00188-04079*-00001-04095
2700 *&02850-00007*&02820-00277*&02788-00513*&02754-00774*&02724-01007*&02686-01257
2800 *&02643-01520*&02593-01757*&02539-02000*&02468-02238*&02370-02504*&02250-02740
2900 *&02083-02995*&01872-03231*&01719-03374*&01581-03478*&01418-03588*&01242-03693
3000 *&00983-03822*&00738-03924*&00506-04006*&00252-04089*-00004-04090
3100 *&03738-00003*&03715-00276*&03688-00515*&03668-00783*&03647-0[illegible]08*&03622-01257
3200 *&03595-01514*&03556-01760*&03508-02001*&03438-02240*&03335-02504*&03201-02741
3300 *&03023-03002*&02793-03244*&02638-03370*&02463-03487*&02219-03616*&01967-03729
3400 *&01718-03825*&01478-03897*&01243-03952*&00988-04012*&00741-04049*&00502-04080
3500 *&00251-04091*-00004-04089
3600 *&04407-00002*&04400-00281*&04391-00529*&04383-00785*&04374-01015*&04365-01265
3700 *&04350-01525*&04335-01770*&04302-02002*&04243-02249*&04162-02505*&04048-02749
3800 *&03886-03005*&03699-03218*&03459-03421*&03200-03581*&02955-03698*&02717-03784
3900 *&02463-03863*&02221-03928*&01961-03976*&01718-04015*&01480-04046*&01243-04072
4000 *&00987-04091*&00739-04097*&00503-04095*&00251-04096*&00001-04093
4100 *&04800-00000*&04797-00282*&04798-00521*&04794-00777*&04798-01009*&04802-01260
4200 *&04798-01522*&04793-01769*&04776-02001*&04744-02248*&04690-02506*&04613-02751
4300 *&04495-03006*&04337-03247*&04201-03393*&04084-03490*&03946-03598*&03695-03728
4400 *&03460-03821*&03199-03886*&02953-03943*&02651-03982*&02455-04010*&02215-04032
4500 *&01963-04051*&01719-04066*&01475-04082*&01244-04093*&00988-04093*&00745-04095
```

TABLE 1 - (continued)

```
4600 *&00501-04093*&00257-04096*-00007-04098
4700 *&04954-00008*&04955-00285*&04959-00777*&04958-01267*&04960-01768*&04927-02249
4800 *&04867-02751*&04757-03139*&04578-03462*&04310-03705*&03944-03888*&03568-03986
4900 *&03204-04031*&02714-04067*&02219-04085*&01715-04092*&01239-04090*&00741-04092
5000 *&00253-04094*-00008-04089
5100 *&04954-00004*&04956-00286*&04956-00781*&04959-01264*&04956-01770*&04959-02249
5200 *&04954-02752*&04902-03250*&04683-03665*&04434-03872*&04193-03993*&03939-04062
5300 *&03460-04072*&02954-04078*&02461-04083*&01964-04090*&01479-04087*&00990-04089
5400 *&00499-04090*&00250-04093*-00008-04091
5500 *&04957-00004*&04953-00286*&04961-00776*&04956-01258*&04962-01768*&04957-02249
5600 *&04954-02754*&04935-03249*&04752-03646*&04459-03917*&04158-04041*&03702-04071
5700 *&03201-04073*&02714-04077*&02220-04084*&01714-04090*&01242-04088*&00741-04092
5800 *&00503-04092*&00250-04095*-00003-04091
5900 *&04957-00004*&04953-00286*&04961-00776*&04956-01258*&04962-01768*&04957-02249
6000 *&04954-02754*&04935-03249*&04752-03646*&04459-03917*&04158-04041*&03702-04071
6100 *&03201-04073*&02714-04077*&02220-04084*&01714-04090*&01242-04088*&00741-04092
6200 *&00503-04092*&00250-04095*-00003-04091
6300 *&04957-00004*&04953-00286*&04961-00776*&04956-01258*&04962-01768*&04957-02249
6400 *&04954-02754*&04935-03249*&04752-03646*&04459-03917*&04158-04041*&03702-04071
6500 *&03201-04073*&02714-04077*&02220-04084*&01714-04090*&01242-04088*&00741-04092
6600 *&00503-04092*&00250-04095*-00003-04091
6700 *-04962&00003*-04961-00245*-04974-00735*-04967-01237*-04963-01719*-04964-02209
6800 *-04965-02717*-04932-03107*-04820-03440*-04658-03682*-04471-03849*-04238-03982
6900 *-03925-04059*-03526-04064*-03003-04065*-02495-04072*-01988-04081*-01500-04089
7000 *-00976-04085*-00491-04091*-00243-04092*-00009-04091
7100 *-04963-00000*-04964-00240*-04970-00734*-04963-01225*-04960-01720*-04933-02190
7200 *-04868-02679*-04777-03074*-04619-03423*-04439-03650*-04205-03818*-03750-04003
7300 *-03363-04055*-02996-04070*-02507-04077*-01988-04082*-01499-04082*-00975-04089
7400 *-00498-04089*-00249-04087*-00008-04089
7500 *-04964&00001*-04953-00241*-04916-00736*-04874-01232*-04818-01720*-04731-02215
7600 *-04591-02717*-04396-03115*-04152-03445*-03848-03688*-03486-03861*-02994-03991
7700 *-02504-04062*-01993-04079*-01495-04085*-00977-04084*-00498-04091*-00241-04094
7800 *-00006-04088
7900 *-04771-00000*-04744-00243*-04669-00739*-04573-01234*-04470-01654*-04333-02084
8000 *-04175-02473*-03974-02843*-03724-03189*-03441-03456*-03065-03700*-02678-03867
8100 *-02267-03984*-01745-04068*-01253-04087*-00748-04093*-00253-04093*-00002-04088
8200 *-04424-00004*-04345-00250*-04165-00743*-03975-01241*-03765-01729*-03527-02226
8300 *-03237-02717*-03014-03023*-02799-03263*-02503-03526*-02161-03741*-01742-03916
8400 *-01348-04020*-00941-04082*-00502-04090*-00209-04091*-00008-04092
8500 *-03849-00001*-03691-00250*-03479-00595*-03260-00939*-03040-01302*-02755-01764
8600 *-02518-02162*-02335-02490*-02140-02834*-01945-03171*-01712-03503*-01353-03813
8700 *-00958-03994*-00548-04088*-00254-04093*-00009-04091
8800 *-03087-00001*-02881-00212*-02640-00466*-02415-00698*-02143-00995*-01945-01247
8900 *-01747-01530*-01567-01845*-01402-02214*-01278-02592*-01168-02973*-01081-03344
9000 *-00981-03594*-00841-03814*-00660-03974*-00428-04083*-00211-04089*-00008-04092
```

TABLE 1 - (continued)

```
9100 *-02616-00001*-02474-00122*-02324-00255*-02107-00438*-01785-00742*-01540-00998
9200 *-01317-01306*-01161-01577*-01059-01843*-00949-02190*-00884-02487*-00851-02741
9300 *-00819-02988*-00788-03224*-00740-03476*-00677-03713*-00558-03949*-00362-04085
9400 *-00145-04091*-00012-04095
9500 *-02042-00006*-01935-00089*-01741-00248*-01497-00447*-01261-00668*-01071-00858
9600 *-00863-01130*-00677-01446*-00591-01689*-00539-01952*-00511-02237*-00496-02490
9700 *-00494-02737*-00496-02984*-00496-03233*-00494-03477*-00470-03711*-00401-03923
9800 *-00258-04067*-00096-04092*-00007-04090
9900 *-01484&00002*-01356-00088*-01254-00167*-01124-00258*-01001-00366*-00835-00500
10000 *-00655-00670*-00529-00816*-00414-01000*-00311-01250*-00260-01502*-00230-01753
10100 *-00214-01991*-00207-02238*-00206-02488*-00197-02739*-00199-02989*-00189-03228
10200 *-00188-03476*-00183-03716*-00179-03961*-00151-04092*-00008-04075
10300 *-00007&00000*-00004-00511*-00005-01004*-00004-01506*-00006-01989*-00006-02491
10400 *-00005-02977*-00011-03480*-00010-04095
10500   -1.0      -0.95     -0.9      -0.85     -0.8      -0.7      -0.6
10600   -0.5      -0.4      -0.3      -0.2      -0.1       0.0       0.1
10700    0.2       0.3       0.4       0.5       0.6       0.7       0.8
10800    0.85      0.9       0.95      1.0
10900    0.34641  -0.27440   0.00758   0.00463   0.00198   0.00211   0.00032
11000    0.44878  -0.29447   0.01009   0.00223   0.00258   0.00369   0.00050
11100    0.52025  -0.27080   0.01227  -0.00004   0.00391   0.00368   0.00081
11200    0.58204  -0.22798   0.01064  -0.00164   0.00543   0.00172   0.00060
11300    0.70588  -0.12377  -0.01246  -0.00275   0.00587   0.00137   0.00117
11400    0.82395  -0.03395  -0.03654  -0.00304   0.00176   0.00157   0.00080
11500    0.92477   0.03544  -0.06407  -0.00478  -0.00266   0.00103  -0.00015
11600    0.99266   0.07721  -0.09170  -0.00561  -0.00276  -0.00075  -0.00013
11700    1.03039   0.09126  -0.11683  -0.00364  -0.00079  -0.00033  -0.00007
11800    1.04960   0.09273  -0.14022  -0.00500   0.00252  -0.00063   0.00099
11900    1.05300   0.09338  -0.14479  -0.00569   0.00431  -0.00028   0.00067
12000    1.05300   0.09338  -0.14479  -0.00569   0.00431  -0.00028   0.00067
12100    1.05300   0.09338  -0.14479  -0.00569   0.00431  -0.00028   0.00067
12200    1.04944   0.09401  -0.13950  -0.00489   0.00276  -0.00121   0.00100
12300    1.03303   0.08996  -0.11985  -0.00025   0.00009  -0.00150   0.00033
12400    0.99788   0.07939  -0.08535   0.00724   0.00040   0.00129   0.00069
12500    0.93959   0.05219  -0.04667   0.01319   0.00088   0.00356   0.00033
12600    0.85550   0.00524  -0.00262   0.02280   0.00562   0.00547   0.00101
12700    0.75006  -0.06606   0.04397   0.03767   0.00606   0.00397   0.00128
12800    0.63970  -0.15695   0.07969   0.05623   0.00548  -0.00072  -0.00031
12900    0.58626  -0.21333   0.08316   0.06482   0.01088  -0.00076  -0.00297
13000    0.53095  -0.27523   0.07389   0.06886   0.01946  -0.00034  -0.00540
13100    0.47381  -0.33047   0.06755   0.05733   0.02485   0.00962  -0.00313
13200    0.41167  -0.41260   0.00234
```

On Table 1, line 100 is simply an alpha numeric title to identify the data in subsequent runs; this title is carried throughout the calculations.

Line 200: IST1 is the number of stations at which offsets are given on the forebody, if the ship is being cut into a fore and afterbody or a parallel middle body is being used. It is the total number of stations, if the ship is being represented by a single polynominal. IST2 is the number of stations on the afterbody; it is zero if there is no separate afterbody representation. KP1 is the degree minus one of the polynominal in S for the forebody or for the full length of the ship if IST2=0. KP2 is the degree minus one of the polynomial for the afterbody or zero if IST2=0.

Line 300: The first number is K, the total number of stations at which offsets are given. N is the number of parameters in the mapping of the unit circle onto the cross section; it should be chosen to be of the order of 1/2 to 2/3 the number of offsets at each of the stations. In the present case N=7 is between 1/3 to 1/2 the number of offsets at the different stations. TOL is the tolerance to which an interaction is solved. BL is the beam-to-length ratio.

Lines 400-500: IFS(I) determines the type of constraints to be used at station I.

IFS(I) = 1. No constraints
2. Surface at station I passes through offsets
3. Slope with respect to S is zero
4. Combination of 2 and 3
5. Slope and curvature with respect to S is zero
6. Combination of 2, 3 and 5

Lines 600-700: JFS(I) is zero if AN is to be read at station I; it is 1 if AN is to be computed.

Lines 900-1100: KT(I) is the number of offsets at station I.

Lines 1300-10400: IX(J) is the x-coordinate of the j-th offset at station I in integer form, as would be obtained from a digitizer. IY(J) is the corresponding Y-coordinate.

Lines 10500-10800: S(I) is the value of S at station I.

Lines 10900-13200: AN(I,J) is the jth value of AN at station I grouped by stations. J=1,...,N.

MAP22 has three output files:

File BNS Table 2 contains the hull geometry matrix.

Line 100: N KPK BL

Lines 200-1500: BN(I,J) grouped by values of J=1, 2, ..., N.

I = 1, ..., KPK

File ANS Table 3, contains starting values of AN for additional calculations.

Lines 100-2600: AN(I,J) I=1, KPK, J=1,... , N grouped by values of I.

File PRINTER is shown as File OPT in Table 4. It contains the data as follows:

Line 100: Ignore the 1 in this line

Line 200: Title

Line 300: "The Input Offsets,", this is a heading for several pages of input effects. In the present example, the input effects ran from lines900 through 62500; Table 4 shows output for stations S(I) = -1.00 S(2) = -0.95 and S(25) = 1.00. XMS is the x-coordinate and YMS is the y-coordinate.

Lines 62900-63500: show the progression of the calculation for AN(I,J), which is typical of the iterative calculations of AN.

Line 63600: In this case, the iteration is not convergent from fourth to fifth iteration, so the results for the fourth iteration are used.

Line 63900: Is typical of lines 63900 through 70800. The AN's for stations 2 through 25 are given in this example and line 63900 gives the values of AN at station I = 2.

Line 71100: MP, KPK, and IT

Lines 71500-72400: Gives intermediate values of a matrix BN used in the computation.

Lines 72700-7400: Gives the intermediate matrix BS

Lines 74300-75900: "The Polynomial Coefficients," these are the final values of the matrix BN(I,J).

Lines 76200-85500: are typical of the computed offsets at the different stations.

Lines 201300-203900: are a printed version of the matrix A(I,J).

For the entire ship starting with just the offsets MAP22 requires about ten minutes of CPU time on the Burroughs B7700; however, the example runs in less than a minute. It is imperative that the input offsets be compared with the output off-

TABLE 2 - HULL GEOMETRY MATRIX

100	7 11	1.00000					
200	1.05407	0.00946	-0.06386	-0.17969	-2.17726	1.30080	1.75289
300	-2.35902	0.99482	1.35102	-1.27218			
400	0.09199	-0.01678	0.12962	0.23842	-1.63679	-0.65025	1.62619
500	1.30540	-1.17223	-0.99895	0.67261			
600	-0.14817	-0.01482	0.27346	0.19518	0.85676	-0.34804	-2.45327
700	0.50034	2.46250	-0.32998	-0.98946			
800	-0.00560	-0.00706	0.00830	0.22146	0.35303	-0.73910	-1.28967
900	1.22848	1.99487	-0.70358	-1.06058			
1000	0.00609	-0.01300	-0.15698	0.25859	1.06743	-1.09828	-2.79267
1100	1.64747	3.20478	-0.79404	-1.32815			
1200	0.00030	-0.01009	-0.05990	0.14915	0.62418	-0.50594	-1.88688
1300	0.62018	2.24264	-0.25234	-0.92001			
1400	0.00096	0.00142	-0.01278	-0.01756	0.05647	0.09382	-0.03964
1500	-0.17823	-0.08303	0.10064	0.07818			

TABLE 3 - FILE ANS

100	0.16513	-0.16512	-0.00003	0.00001	0.00001	-0.00001	0.00000
200	0.16513	-0.16512	-0.00003	0.00001	0.00001	-0.00001	0.00000
300	0.34641	-0.27440	0.00758	0.00463	0.00198	0.00211	0.00032
400	0.44878	-0.29447	0.01009	0.00223	0.00258	0.00369	0.00050
500	0.52025	-0.27080	0.01227	-0.00004	0.00391	0.00368	0.00081
600	0.58204	-0.22798	0.01064	-0.00164	0.00543	0.00172	0.00060
700	0.70588	-0.12377	-0.01246	-0.00275	0.00587	0.00137	0.00117
800	0.82395	-0.03395	-0.03654	-0.00304	0.00176	0.00157	0.00080
900	0.92477	0.03544	-0.06407	-0.00478	-0.00266	0.00103	-0.00015
1000	0.99266	0.07721	-0.09170	-0.00561	-0.00276	-0.00075	-0.00013
1100	1.03039	0.09126	-0.11683	-0.00364	-0.00079	-0.00033	-0.00007
1200	1.04960	0.09273	-0.14022	-0.00500	0.00252	-0.00063	0.00099
1300	1.05300	0.09338	-0.14479	-0.00569	0.00431	-0.00028	0.00067
1400	1.05300	0.09338	-0.14479	-0.00569	0.00431	-0.00028	0.00067
1500	1.05300	0.09338	-0.14479	-0.00569	0.00431	-0.00028	0.00067
1600	1.04944	0.09401	-0.13950	-0.00489	0.00276	-0.00121	0.00100
1700	1.03303	0.08996	-0.11985	-0.00025	0.00009	-0.00150	0.00033
1800	0.99788	0.07989	-0.08535	0.00724	0.00040	0.00129	0.00069
1900	0.93959	0.05219	-0.04667	0.01319	0.00088	0.00356	0.00033
2000	0.85550	0.00524	-0.00262	0.02280	0.00562	0.00547	0.00101
2100	0.75006	-0.06606	0.04397	0.03767	0.00606	0.00397	0.00128
2200	0.63970	-0.15695	0.07969	0.05623	0.00548	-0.00072	-0.00031
2300	0.58626	-0.21333	0.08316	0.06482	0.01088	-0.00076	-0.00297
2400	0.53095	-0.27523	0.07389	0.06886	0.01946	-0.00034	-0.00540
2500	0.47381	-0.33047	0.06755	0.05733	0.02485	0.00962	-0.00313
2600	0.41167	-0.41260	0.00234	0.00000	0.00000	0.00000	0.00000

TABLE 4 - PRINTER FILE FOR MAP22

```
100 1
200          SSPA MODEL 720
300   THE INPUT OFFSETS
400
500
600
700
800
900                       XMS              YMS              S( 1)=-1.00
1000
1100
1200                      0.00000          0.00000
1300                      0.00000         -0.10093
1400                      0.00000         -0.20186
1500                      0.00000         -0.33024
1600
1700
1800                       XMS              YMS              S( 2)=-0.95
1900
2000
2100                      0.08862         -0.00101
2200                      0.08216         -0.05491
2300                      0.07691         -0.10315
2400                      0.07045         -0.15462
2500                      0.06681         -0.20226
2600                      0.06237         -0.25192
2700                      0.05793         -0.30339
2800                      0.05329         -0.35386
2900                      0.04885         -0.40210
3000                      0.04380         -0.45135
3100                      0.03694         -0.50343
3200                      0.02866         -0.55369
3300                      0.01958         -0.60174
3400                      0.01292         -0.62333
3500                     -0.00020         -0.62394
```

TABLE 4 - (continued)

```
                    XMS             YMS           S(25)= 1.00


                  0.00141         0.00000
                  0.00081        -0.10315
                  0.00101        -0.20266
                  0.00081        -0.30400
                  0.00121        -0.40149
                  0.00121        -0.50283
                  0.00101        -0.60093
                  0.00222        -0.70246
                  0.00202        -0.82660



 INITIAL AN( 1,J)=  0.00000 -0.16512  0.00000    A-EA COEFFICIENT= -1.00000


 DUT= 0.00000  AN=  0.16512 -0.16512  0.00000
 DUT= 0.00000  AN=  0.16513 -0.16513 -0.00001  0.00001
 DUT= 0.00000  AN=  0.16513 -0.16513 -0.00002  0.00001 -0.00000
 DUT= 0.00000  AN=  0.16513 -0.16512 -0.00003  0.00001  0.00001 -0.00001
  FIT IS GOING BAD  DUT=0.00000


AN GIVEN AT I= 2 AN=  0.34641 -0.27440  0.00758  0.00463  0.00198  0.00211  0.00032

MP=   11 KPK=   11 INT=    0



    THE VECTORS BN


 19.01675   0.60387   6.08532   0.50995   3.72070   0.45222   2.72657   0.41271   2.17005   0.38382   1.81186
 -1.94706  -0.10453  -2.21187  -0.17937  -1.89257  -0.21743  -1.63013  -0.23813  -1.43171  -0.24976  -1.28060
 -0.89903   0.30259   0.17484   0.20837   0.19406   0.15325   0.16108   0.11771   0.13006   0.09322   0.10559
```

TABLE 4 - (continued)

```
    0.28630   0.27041   0.22098   0.18581   0.16476   0.13736   0.12724   0.10580   0.10091   0.08374   0.08163
    0.10726   0.04867   0.06766   0.03778   0.05217   0.03124   0.04160   0.02643   0.03396   0.02262   0.02822
    0.03199   0.00357   0.02218   0.00168   0.01615   0.00151   0.01279   0.00174   0.01062   0.00198   0.00906
   -0.00032  -0.01164  -0.00578  -0.00984  -0.00566  -0.00807  -0.00503  -0.00661  -0.00434  -0.00545  -0.00370


     THE MATRIX BS


   25.00000   0.00000  10.95000  -0.00000   7.73963   0.00000   6.18129  -0.00000   5.22645   0.00000   4.57409
    0.00000  10.95000  -0.00000   7.73963   0.00000   6.18129  -0.00000   5.22645   0.00000   4.57409   0.00000
   10.95000  -0.00000   7.73963   0.00000   6.18129  -0.00000   5.22645   0.00000   4.57409   0.00000   4.10006
   -0.00000   7.73963   0.00000   6.18129  -0.00000   5.22645   0.00000   4.57409   0.00000   4.10006   0.00000
    7.73963   0.00000   6.18129  -0.00000   5.22645   0.00000   4.57409   0.00000   4.10006   0.00000   3.74165
    0.00000   6.18129  -0.00000   5.22645   0.00000   4.57409   0.00000   4.10006   0.00000   3.74165   0.00000
    6.18129  -0.00000   5.22645   0.00000   4.57409   0.00000   4.10006   0.00000   3.74165   0.00000   3.46290
   -0.00000   5.22645   0.00000   4.57409   0.00000   4.10006   0.00000   3.74165   0.00000   3.46290  -0.00000
    5.22645   0.00000   4.57409   0.00000   4.10006   0.00000   3.74165   0.00000   3.46290  -0.00000   3.24141
    0.00000   4.57409   0.00000   4.10006   0.00000   3.74165   0.00000   3.46290  -0.00000   3.24141  -0.00000
    4.57409   0.00000   4.10006   0.00000   3.74165   0.00000   3.46290  -0.00000   3.24141  -0.00000   3.06237


    THE POLYNOMIAL COEFFICIENTS


  J = 1    1.05407   0.00946  -0.06386  -0.17969  -2.17726   1.30080   1.75289  -2.35902   0.99482   1.35102
          -1.27218
  J = 2    0.09199  -0.01678   0.12962   0.23842  -1.63679  -0.65025   1.62619   1.30540  -1.17223  -0.99895
           0.67261
  J = 3   -0.14817  -0.01482   0.27346   0.19518   0.85676  -0.34804  -2.45327   0.50034   2.46250  -0.32998
          -0.98946
  J = 4   -0.00560  -0.00706   0.00830   0.22146   0.35303  -0.73910  -1.28967   1.22848   1.99487  -0.70358
          -1.06058
  J = 5    0.00609  -0.01300  -0.15698   0.25859   1.06743  -1.09828  -2.79267   1.64747   3.20478  -0.79404
          -1.32815
  J = 6    0.00030  -0.01009  -0.05990   0.14915   0.62418  -0.50594  -1.88688   0.62018   2.24264  -0.25234
          -0.92001
  J = 7    0.00096   0.00142  -0.01278  -0.01756   0.05647   0.09382  -0.03964  -0.17823  -0.08303   0.10064
           0.07818
```

TABLE 4 - (continued)

76000			
76100			
76200	X	Y	S(1)=******
76300			
76400	-0.00200	0.00000	
76500	-0.00197	-0.01188	
76600	-0.00187	-0.02373	
76700	-0.00171	-0.03552	
76800	-0.00150	-0.04721	
76900	-0.00126	-0.05879	
77000	-0.00101	-0.07024	
77100	-0.00076	-0.08153	
77200	-0.00054	-0.09267	
77300	-0.00035	-0.10365	
77400	-0.00021	-0.11446	
77500	-0.00013	-0.12512	
77600	-0.00011	-0.13563	
77700	-0.00014	-0.14598	
77800	-0.00022	-0.15620	
77900	-0.00033	-0.16627	
78000	-0.00046	-0.17620	
78100	-0.00059	-0.18597	
78200	-0.00069	-0.19556	
78300	-0.00076	-0.20497	
78400	-0.00077	-0.21415	
78500	-0.00072	-0.22310	
78600	-0.00061	-0.23176	
78700	-0.00043	-0.24011	
78800	-0.00020	-0.24813	
78900	0.00006	-0.25579	
79000	0.00034	-0.26307	
79100	0.00062	-0.26995	
79200	0.00087	-0.27644	
79300	0.00108	-0.28253	
79400	0.00122	-0.28824	
79500	0.00130	-0.29356	
79600	0.00130	-0.29853	
79700	0.00122	-0.30315	
79800	0.00109	-0.30744	
79900	0.00091	-0.31141	
80000	0.00070	-0.31506	
80100	0.00048	-0.31839	
80200	0.00028	-0.32140	
80300	0.00010	-0.32406	
80400	-0.00003	-0.32637	
80500	-0.00011	-0.32829	
80600	-0.00013	-0.32982	
80700	-0.00012	-0.33092	
80800	-0.00007	-0.33159	
80900	0.00000	-0.33181	

TABLE 4 - (continued)

81100			
81200	X	Y	S(2)=-.9500
81300			
81400	0.09246	0.00000	
81500	0.09204	-0.01806	
81600	0.09081	-0.03628	
81700	0.08885	-0.05481	
81800	0.08627	-0.07376	
81900	0.08322	-0.09321	
82000	0.07989	-0.11319	
82100	0.07644	-0.13368	
82200	0.07304	-0.15461	
82300	0.06983	-0.17587	
82400	0.06693	-0.19732	
82500	0.06441	-0.21880	
82600	0.06229	-0.24013	
82700	0.06055	-0.26115	
82800	0.05914	-0.28172	
82900	0.05797	-0.30170	
83000	0.05692	-0.32102	
83100	0.05589	-0.33965	
83200	0.05477	-0.35756	
83300	0.05346	-0.37480	
83400	0.05189	-0.39141	
83500	0.05003	-0.40747	
83600	0.04787	-0.42305	
83700	0.04544	-0.43823	
83800	0.04280	-0.45306	
83900	0.04003	-0.46758	
84000	0.03721	-0.48180	
84100	0.03444	-0.49568	
84200	0.03182	-0.50918	
84300	0.02940	-0.52222	
84400	0.02723	-0.53470	
84500	0.02534	-0.54653	
84600	0.02372	-0.55759	
84700	0.02232	-0.56780	
84800	0.02109	-0.57707	
84900	0.01996	-0.58535	
85000	0.01883	-0.59261	
85100	0.01761	-0.59886	
85200	0.01624	-0.60412	
85300	0.01464	-0.60845	
85400	0.01279	-0.61192	
85500	0.01065	-0.61461	

TABLE 4 - (continued)

AN(I.J)

0.16591	-0.16644	-0.00086	0.00016	-0.00022	-0.00062	0.00007
0.34432	-0.27111	0.00894	0.00412	0.00232	0.00372	0.00016
0.44933	-0.29486	0.01181	0.00264	0.00296	0.00338	0.00049
0.52158	-0.27320	0.01079	0.00015	0.00375	0.00229	0.00030
0.58279	-0.22889	0.00689	-0.00171	0.00479	0.00177	0.00097
0.70380	-0.12268	-0.00905	-0.00303	0.00550	0.00217	0.00089
0.82412	-0.03057	-0.03465	-0.00336	0.00265	0.00206	0.00053
0.92493	0.03424	-0.06531	-0.00413	-0.00167	0.00058	0.00017
0.99372	0.07303	-0.09475	-0.00493	-0.00377	-0.00094	0.00001
1.03136	0.09142	-0.11833	-0.00513	-0.00198	-0.00117	0.00009
1.04730	0.09630	-0.13450	-0.00492	0.00221	-0.00023	0.00036
1.05244	0.09457	-0.14406	-0.00499	0.00568	0.00063	0.00071
1.05407	0.09199	-0.14817	-0.00560	0.00609	0.00030	0.00096
1.05399	0.09168	-0.14664	-0.00597	0.00358	-0.00110	0.00096
1.04899	0.09302	-0.13752	-0.00464	0.00049	-0.00219	0.00070
1.03268	0.09160	-0.11817	-0.00049	-0.00046	-0.00137	0.00037
0.99791	0.08055	-0.08729	0.00629	0.00142	0.00162	0.00029
0.93919	0.05292	-0.04656	0.01442	0.00397	0.00485	0.00067
0.85530	0.00412	-0.00089	0.02349	0.00419	0.00534	0.00128
0.75152	-0.06681	0.04288	0.03612	0.00279	0.00207	0.00113
0.63963	-0.15837	0.07622	0.05573	0.00687	-0.00084	-0.00100
0.58464	-0.21171	0.08440	0.06598	0.01310	0.00008	-0.00276
0.53079	-0.27067	0.08160	0.06994	0.02025	0.00289	-0.00427
0.47541	-0.33642	0.05972	0.05556	0.02147	0.00543	-0.00414
0.41105	-0.41076	0.00449	0.00054	0.00126	0.00128	0.00025

sets and different values of KPK are tried in order to get the smoothest representation in the lengthwise direction. Three or four runs should be sufficient to develop a good choice of KPK.

3.2 TBL18

The primary output from this program is the boundary-layer parameters obtained using the small cross-flow assumption. Input data is entered either through the keyboard in answer to prompting from the computer, or it is read from one of two stored files: File 8 TITLE = "BNSAL" is the BNS file created by MAP22; FILE 10 TITLE = "TBLINP" is a file created by the present program during a previous run.

The example form of TBLINP is shown in Table 5.

Line 100: 40 is the free stream velocity U

Line 200: is the beam-to-length ratio BL

Line 300: NSTAT the number of stations at which the boundary layer is calculated.

Line 400: S1, S2 the first and last stations in the calculations.

Line 500: NTH, the number of streamlines being used.

Lines 600-2100: THATA values of the streamlines at station S1.

Lines 2200-3700: initial values of H, θ_{11}, tan β for each value of THATA.

Line 3800: Reynolds number RE

File BNS is read first and is printed at the terminal, so a quick glance can determine if the correct file was loaded. Table 6 shows the terminal display as it should appear after going through all the computer prompts. The computer first prompts you for a choice of input method: Enter 1 if the input data is on cards, 2 if it is to come from the terminal, and 3 if it is to be read from TBLINP. If you choose 1 or 3, you merely wait for the program to finish; if you choose 2, the computer will prompt you to enter U0. It prompts you by displaying the desired variable with an equal sign, some space, and the format; in the case of U0 one gets U0=1F10.5 Simply enter a space and 40.0, and it responds with 40.00000. The computer will then prompt you for the remaining variables BL, NSTAT, S1, etc. These are the variables that are previously described under FILE TBLINP. In running the example for the first time, have Table 5 next to you and simply type the numbers from it.

In general the choice of θ values depends on how the streamlines behave at the other stations. If they converge too closely in a region of interest addi-

TABLE 5 - FILE TBLINP

100	40.00000		
200	0.14150		
300	21		
400	-0.60000	0.60000	
500	16		
600	0.00000		
700	-0.10472		
800	-0.20944		
900	-0.31416		
1000	-0.41888		
1100	-0.52360		
1200	-0.62832		
1300	-0.73304		
1400	-0.83776		
1500	-0.94248		
1600	-0.04720		
1700	-1.15192		
1800	-1.25664		
1900	-1.36136		
2000	-1.46608		
2100	-1.57080		
2200	1.80000	0.00040	0.00000
2300	1.80000	0.00040	0.00000
2400	1.80000	0.00040	0.00000
2500	1.80000	0.00040	0.00000
2600	1.80000	0.00040	0.00000
2700	1.80000	0.00040	0.00000
2800	1.80000	0.00040	0.00000
2900	1.80000	0.00040	0.00000
3000	1.80000	0.00040	0.00000
3100	1.80000	0.00040	0.00000
3200	1.80000	0.00040	0.00000
3300	1.80000	0.00040	0.00000
3400	1.80000	0.00040	0.00000
3500	1.80000	0.00040	0.00000
3600	1.80000	0.00040	0.00000
3700	1.80000	0.00040	0.00000
3800		1600000.0	

TABLE 6 - TERMINAL DISPLAY FOR TBL18

```
          SSPA MODEL 720
   7    11
    1.05407   0.00946  -0.06386  -0.17969  -2.17726   1.30080   1.75289
   -2.35902   0.99482   1.35102  -1.27218
    0.09199  -0.01678   0.12962   0.23842  -1.63679  -0.65025   1.62619
    1.30540  -1.17223  -0.99895   0.67261
   -0.[illegible]17  -0.01482   0.27346   0.19518   0.85676  -0.34804  -2.45327
    0.30034   2.46250  -0.32998  -0.98946
   -0.00560  -0.00706   0.00830   0.22146   0.35303  -0.73910  -1.28967
    1.22848   1.99487  -0.70358  -1.06058
    0.00609  -0.01300  -0.15698   0.25859   1.06743  -1.09828  -2.79267
    1.64747   3.20478  -0.79404  -1.32815
    0.00030  -0.01009  -0.05990   0.14915   0.62418  -0.50594  -1.88688
    0.62018   2.24264  -0.25234  -0.92001
    0.00096   0.00142  -0.01278  -0.01756   0.05647   0.09382  -0.03964
   -0.17823  -0.08303   0.10064   0.07818
NINPT DENOTES THE TYPE OF INPUT
NINPT=1  CARD DATA
      2  REMOTE
      3  STORED DATA


NINPT     1I1
#?
2
UU=       1F10.5
 40.0
 40.00000
BL=       1F10.5
 0.14150
  0.14150
NSTAT=          1I5
   21
  21
    S1        S2         7F10.5
 -0.6      0.6
  -0.60000   0.60000
NTH=     1I5
   16
  16
TTH=          1F10.5
 0.0000
-0.10472
-0.20944
-0.341416
-0.41888
```

TABLE 6 - (continued)

```
-0.52360
-0.62832
-0.73304
-0.83776
-0.9248
-0.04720
-1.15192
-1.25664
-1.36136
-1.46608
-1.57080
 0.00000
-0.10472
-0.20944
-0.34142
-0.41888
-0.52360
-0.62832
-0.73304
-0.83776
-0.92480
-0.04720
-1.15192
-1.25664
-1.36136
-1.46608
-1.57080
 HB          THMB        BETB   3F10.5
1.80000    0.0004        0.0
 1.80000     0.00040     0.00000
 1.8            0.0004  0.0
 1.80000     0.00040     0.00000
 1.8          0.0004      0.0
 1.80000     0.00040     0.00000
 1.8           0.0       0.0
 1.80000     0.00000     0.00000
 1.8         0.0004      0.0
 1.80000     0.00040     0.00000
 1.8         0.0004      0.0
 1.80000     0.00040     0.00000

 1.8         0.0004    0.0
 1.80000     0.00040     0.00000
 1.8         0.0004      0.0
 1.80000     0.00040     0.00000
 1.8         0.0004      0.0
 1.80000     0.00040     0.00000
 1.8         0.0004      0.0
 1.80000     0.00040     0.00000
```

TABLE 6 - (continued)

```
  1.8       0.0004  0.0
  1.80000   0.00040   0.00000
  1.8       0.0004    0.0
  1.80000   0.00040   0.00000
  1.8       0.0004    0.0
  1.80000   0.00040   0.00000
  1.8       0.0004    0.0
  1.80000   0.00040   0.00000
  1.8       0.0004    0.0
  1.80000   0.00040   0.00000
  1.8       0.0004    0.0
  1.80000   0.00040   0.00000
RE=          1F20.1
  1600000.0
         1600000.0
```

tional streamlines should be presented. The initial boundary layer parameters have to be determined experimentally at this time. Interval size should be halved and the solution repeated until it converges, since the integration method is the simple Euler method.

In the process of entering the data from the terminal the computer creates FILE TBLSAVE. This file is exactly FILE TBLINP with a different name; thus, once a set of data has been entered it is saved. If you are nearly finished entering data and an error is made simply continue with the rest of the entries and make editorial corrections to TBLSAVE.

All the computational results are output to the printer. Table 7 shows the output for the example case.

Line 300: INPUT signifies that the lines between 300 and 2300 are the input data.

Line 500: N is the number of parameters used in the mapping, and KPK is the degree plus one of the longitudinal polynomials, which are used to interpret the mapping parameters. Here, as before, BL is the beam-to-length ratio.

Line 600: NTH and NSTAT

Line 700: S1 and S2

Lines 800-2100: present the components of the matrix BN(I,J).

Line 2300: Alpha-numeric Title identifying the data

The remaining data are organized by station and are tabulated in twenty columns each headed by the variable name. Lines 49600-54000 show the example data for station S = -0.0286.

TH	θ coordinate of streamline measured from x-axis
X	x-coordinate of the streamline
Y	y-coordinate of the streamline
CP	pressure coefficient
PSI	$\partial p/\partial S$ rate of change in pressure along streamline
DTDS	$d\theta/ds$
R	metric along the streamline
K1	geodesic curvature of the streamlines
K2	geodesic curvature of the orthogonal trajectories to the streamlines
PHI	potential at (S,θ)

TABLE 7 - PRINTER FILE FOR TBL18

```
300   INPUT
400
500   N=  7 KPK= 11 BL=    0.14150
600   NTH=    16 NSTAT=    21
700   S1=  -0.60000 S2=    0.60000
800   .10541E+01      .94600E-02      -.63860E-01     -.17969E+00     -.21773E+01     .13008E+01      .17529E+01
900  -.23590E+01      .99482E+00      .13510E+01      -.12722E+01
1000  .91990E-01     -.16780E-01      .12962E+00      .23842E+00      -.16368E+01     -.65025E+00     .16262E+01
1100  .13054E+01     -.11722E+01      -.99895E+00     .67261E+00
1200 -.14817E+00     -.14820E-01      .27346E+00      .19518E+00      .85676E+00      -.34804E+00     -.24533E+01
1300  .50034E+00      .24625E+01      -.32998E+00     -.98946E+00
1400 -.56000E-02     -.70600E-02      .83000E-02      .22146E+00      .35303E+00      -.73910E+00     -.12897E+01
1500  .12285E+01      .19949E+01      -.70358E+00     -.10606E+01
1600  .60900E-02     -.13000E-01      -.15698E+00     .25859E+00      .10674E+01      -.10983E+01     -.27927E+01
1700  .16475E+01      .32048E+01      -.79404E+00     -.13282E+01
1800  .30000E-03     -.10090E-01      -.59900E-01     .14915E+00      .62418E+00      -.50594E+00     -.18869E+01
1900  .62018E+00      .22426E+01      -.25234E+00     -.92001E+00
2000  .96000E-03      .14200E-02      -.12780E-01     -.17560E-01     .56470E-01      .93820E-01      -.39640E-01
2100 -.17823E+00     -.83030E-01      .10064E+00      .78180E-01
2200 1
2300          SSPA MODEL 720
```

S(1)= -0.6000

TH	X	Y	CP	PSI	DTDS	R	K1	K2	PHI
0.00000	0.10765	0.00000	-0.08005	0.19940	0.00000	.16232E+00	-1.43932	0.00000	-0.62943
-0.10472	0.10708	-0.01384	-0.07820	0.19875	0.01425	.19694E+00	-1.69282	0.12009	-0.62941
-0.20944	0.10553	-0.02773	-0.07328	0.18720	-0.06405	.30765E+00	-2.29563	0.19432	-0.62927
-0.31416	0.10334	-0.04144	-0.06770	0.15529	-0.27847	.34088E+00	-3.11100	0.15728	-0.62882
-0.41888	0.10069	-0.05443	-0.06567	0.[illegible]1814	-0.66294	.17911E+00	-4.18559	-0.04878	-0.62796
-0.52360	0.09740	-0.06613	-0.07158	0.10961	-1.25183	.85478E-01	-5.27692	-0.43345	-0.62674
-0.62832	0.09301	-0.07623	-0.08605	0.11366	-1.95074	.54866E-01	-5.24733	-0.72843	-0.62530
-0.73304	0.08715	-0.08483	-0.10151	0.03223	-2.44676	.53001E-01	-3.59550	-0.55591	-0.62380
-0.83776	0.07970	-0.09220	-0.11058	-0.07266	-2.59049	.68376E-01	-1.98712	-0.24503	-0.62230
-0.94248	0.07081	-0.09859	-0.11414	-0.12638	-2.53445	.91207E-01	-1.09771	-0.06308	-0.62079
-0.04720	0.10753	-0.00623	-0.07967	0.19944	0.01302	.16906E+00	-1.49415	0.05617	-0.62943
-1.15192	0.04975	-0.10859	-0.10768	-0.23432	-2.13829	.12256E+00	0.40676	0.34074	-0.61782
-1.25664	0.03800	-0.11212	-0.09556	-0.27278	-1.76437	.13780E+00	1.25993	0.53960	-0.61649
-1.36136	0.02566	-0.11465	-0.07968	-0.24626	-1.26584	.15333E+00	2.04753	0.58412	-0.[illegible]

S(11)= -0.0286

TABLE 7 - (continued)

TH	X	Y	CP	PSI	DTUS	R	K1	K2	PHI
0.00000	0.14167	0.00000	-0.06067	-0.07010	0.00000	.18083E+01	0.15363	0.00000	-0.02928
-0.08270	0.14160	-0.01603	-0.06029	-0.06811	0.02576	.28599E+01	0.14098	0.02108	-0.02930
-0.17462	0.14143	-0.03362	-0.05934	-0.06232	0.05063	.73034E+01	0.10569	0.02549	-0.02936
-0.28409	0.14129	-0.05373	-0.05881	-0.05392	0.07320	.35798E+01	0.05935	-0.00799	-0.02947
-0.44192	0.14109	-0.07917	-0.06166	-0.04383	0.09961	.27284E+00	0.03128	-0.10535	-0.02965
-0.97222	0.11231	-0.11589	-0.07080	-0.00513	-0.00392	.10558E+00	-0.26025	0.03616	-0.03004
-1.16073	0.08351	-0.11658	-0.06923	0.00428	-0.02026	.26076E+01	-0.00244	0.01301	-0.03006
-1.24728	0.06741	-0.11651	-0.06898	0.00788	-0.01138	.25394E+01	0.02966	0.00230	-0.03006
-1.30579	0.05581	-0.11651	-0.06898	0.01010	-0.00481	.22215E+01	0.02937	-0.00247	-0.03006
-1.35367	0.04601	-0.11655	-0.06906	0.01179	-0.00040	.30584E+01	0.01927	-0.00468	-0.03007
-0.03668	0.14165	-0.00712	-0.06060	-0.06970	0.01164	.19587E+01	0.15104	0.01039	-0.02928
-1.43678	0.02859	-0.11669	-0.06926	0.01429	0.00351	.18340E+02	-0.00955	-0.00511	-0.03008
-1.47371	0.02074	-0.11676	-0.06933	0.01513	0.00352	.44627E+01	-0.02247	-0.00419	-0.03009
-1.50802	0.01343	-0.11681	-0.06939	0.01570	0.00272	.28622E+01	-0.03212	-0.00291	-0.03009
-1.54011	0.00657	-0.11683	-0.06942	0.01603	0.00145	.23597E+01	-0.03788	-0.00148	-0.03010
-1.57080	-0.00000	-0.11684	-0.06943	0.01613	-0.00000	.22320E+01	-0.03977	0.00000	-0.03010

BOUNDARY LAYER AT S(11)= -0.0286

H	TH //	DE	BETA	CF	RS	RTH	CFW	CFB	CFS
1.48872	0.00163	0.00242	0.00000	3.61366	0.	.1341E+04	0.00014	3.61366	3.61365
1.48933	0.00162	0.00242	0.01136	3.61365	0.	.1337E+04	-0.04090	3.61365	3.61364
1.49091	0.00161	0.00240	0.02372	3.61478	0.	.1327E+04	-0.08559	3.61478	3.61477
1.49305	0.00157	0.00234	0.03282	3.63944	0.	.1291E+04	-0.11930	3.63944	3.63942
1.49815	0.00133	0.00200	0.03242	3.83824	0.	.1099E+04	-0.12429	3.83824	3.83823
1.49227	0.00113	0.00169	0.03145	4.12243	0.	.9367E+03	-0.12948	4.12243	4.12241
1.47315	0.00150	0.00221	0.04327	3.80806	0.	.1242E+04	-0.16464	3.80806	3.80805
1.46550	0.00165	0.00241	0.05156	3.72506	0.	.1362E+04	-0.19192	3.72506	3.72505
1.46108	0.00173	0.00252	0.05715	3.68684	0.	.1428E+04	-0.21054	3.68684	3.68683
1.45820	0.00177	0.00258	0.05824	3.67360	0.	.1461E+04	-0.21380	3.67360	3.67359
1.48885	0.00163	0.00242	0.00499	3.61367	0.	.1340E+04	-0.01789	3.61367	3.61366
1.45421	0.00180	0.00262	0.05075	3.67255	0.	.1488E+04	-0.18622	3.67255	3.67254
1.45277	0.00181	0.00262	0.04277	3.67585	0.	.1494E+04	-0.15705	3.67585	3.67584
1.45178	0.00181	0.00262	0.03145	3.68022	0.	.1495E+04	-0.11560	3.68022	3.68020
1.45122	0.00181	0.00262	0.01684	3.68400	0.	.1494E+04	-0.06191	3.68400	3.68398
1.45105	0.00181	0.00262	-0.00000	3.68551	0.	.1494E+04	0.00015	3.68551	3.68549

H	shape parameter
TH11	θ_{11}
DE	$H\ \theta_{11}$
BETA	β
CF	coefficient of skin friction
RS	not used
RTH	local Reynolds number
CFW	skin-friction coefficient in normal direction
CFB	skin-friction coefficient in crossflow direction
CFS	skin-friction coefficient in streamwise direction

3.3 TBL3D

The primary output from TBL3D are the surface coordinates, distance metrics, and geodesic curvatures. The actual operation of the program is very similar to the operation of TBL18. Input data is entered either through the keyboard in response to the computer's prompting or it is entered from one of two stored files: FILE 8 TITLE = "BNSAL" is the BNS file created by MAP 22, FILE 10 TITLE = "TBL3DINP" is a file created by the present program during a previous run.

The example form of TBL3DINP is shown in Table 8.

TABLE 8 - FILE TBL3DINP

100	0.14150
200	-0.60000
300	0.60000
400	21
500	16

Table 8: File TBL3DINP

Line 100: BL

Line 200: START value of S at initial station

Line 300: SFINIS value of S at final station

Line 400: NSTAT number of stations

Line 500: NTH number of θ-values used

There is no reason to enter the θ-values as they are equally spaced from 0 to $-\pi/2$.

Since one does not have to enter θ or the initial boundary layer parameters,

the initial run with this program is simpler than it is with TBL18. Table 9 shows the terminal display as it would appear for TBL3D after going through an initial run with it. Again the same menu for NINPT. One then enters a 1, if a BL other than 1 is to be used followed by BY, START, SFINIS, NSTAT, and NTH. After the computer responds with the value of NTH, 16 in this case, it prints the NTH values of θ.

In general, several values of NTH should be tried such as 20, 30, 40, and 50 to check the accuracy of the final calculation. The number of stations should be chosen experimentally after examining the results from TBLSOL. Making several runs with NTH = 20 and NSTAT chosen so that ΔS = 0.050, 0.025, 0.020, and 0.010 should provide a good feel for how the solution behaves with respect to ΔS. If some area needs further refinement of the grid in the s-direction for convergence, the program can be run just for this area with appropriate choices of START and SFINIS.

The output file TBLSAVE is created in the process of entering the data from the terminal; it is exactly TBL3DINP with a different name.

All the computational results are saved in TBL3DOPT and also an identical copy on the PRINTER file. Table 10 shows typical output.

Line 100: identifies the data

Line 200: N, KPK, BL

Line 300: START, SFINIS, NSTAT, NTH

Line 400: Station number, value of S at the station

Except for lines 400, 5300, and 6200, lines beginning with 1, 2, 3 have the following interpretation:

1	x	y	θ	u	v
2	w	u_ϕ	u_θ	u_n	S
3	K_ϕ	K_θ	H	F	G

The values (x, y, θ, S) are the respective coordinates of the grid point. u, v, w, are the inviscid velocities of the fluid in the (x, y, s) coordinate system; u_ψ and u_θ are the velocities in the ϕ and θ-directions respectively, while u_n is a normal velocity to the surface. Since the inviscid velocities are obtained here with slender body theory, u_n is not identically zero. K_ϕ and K_θ are the respective geodesic curvatures. H, F, and G are functions used in computing the metrics of the surface coordinates and are defined in Equations (22) and (23).

TABLE 9 - TERMINAL DISPLAY FOR TBL3D

```
          SSPA MODEL 720
   7   11    1.00000
  1.05407    0.00946   -0.06386   -0.17969   -2.17726    1.30080    1.75289
 -2.35902    0.99482    1.35102   -1.27218
  0.09199   -0.01678    0.12962    0.23842   -1.63679   -0.65025    1.62619
  1.30540   -1.17223   -0.99895    0.67261
 -0.14817   -0.01482    0.27346    0.19518    0.85676   -0.34804   -2.45327
  0.50034    2.46250   -0.32998   -0.98946
 -0.00560   -0.00706    0.00830    0.22146    0.35303   -0.73910   -1.28967
  1.22848    1.99487   -0.70358   -1.06058
  0.00609   -0.01300   -0.15698    0.25859    1.06743   -1.09828   -2.79267
  1.64747    3.20478   -0.79404   -1.32815
  0.00030   -0.01009   -0.05990    0.14915    0.62418   -0.50594   -1.88688
  0.62018    2.24264   -0.25234   -0.92001
  0.00096    0.00142   -0.01278   -0.01756    0.05647    0.09382   -0.03964
 -0.17823   -0.08303    0.10064    0.07818
NINPT DENOTES THE TYPE OF INPUT
NINPT=1  CARD DATA
      2  REMOTE
      3  STORED DATA


NINPT     1I1
#?
2
DO YOU WISH TO ENTER NEW BL?
IF YES TYPE 1 NO TYPE 0
1
BL=    1F10.5
  0.1415
  0.14150
START=      1F10.5
  -0.6
 -0.60000
SFINIS=      1F10.5
   0.6
  0.60000
NSTAT=       1I3
 21
  21
NTH=         1I3
 16
  16
  0.00000  -0.10472  -0.20944  -0.31416  -0.41888  -0.52360  -0.62832
 -0.73304  -0.83776  -0.94248  -1.04720  -1.15192  -1.25664  -1.36136
-1.46608  -1.57080
```

TABLE 10 - FILE TBL3DOPT

```
100          SSPA MODEL 720
200          7          11   0.14150
300    -0.60000   0.60000         21          16
400    1-0.600000
500    1    .1076487E+00  0.             0.              .2228504E+00  0.
600    2    .1015192E+01   .1039358E+01  0.             -.3304451E-02  -.6000000E+00
700    3   -.1746801E+01  0.              .1024530E+01  0.              .1319586E+00
800    1    .1070765E+00  -.1383975E-01  -.1047198E+00   .2260201E+00  -.2191698E-01
900    2    .1013320E+01   .1038446E+01   .1965947E-02  -.2949163E-02  -.6000375E+00
1000   3   -.1503672E+01  -.1295564E+00   .1025467E+01   .7161124E-03   .1329239E+00
1100   1    .1055012E+00  -.2772837E-01  -.2094395E+00   .2319952E+00  -.5121243E-01
1200   2    .1008394E+01   .1035967E+01  -.8533613E-02  -.1934473E-02  -.6001429E+00
1300   3   -.9778611E+00  -.2079391E+00   .1027713E+01   .1298230E-02   .1337157E+00
1400   1    .1032734E+00  -.4142935E-01  -.3141593E+00   .2340332E+00  -.9026146E-01
1500   2    .1002341E+01   .1032607E+01  -.3647421E-01  -.5599877E-03  -.6002810E+00
1600   3   -.4966123E+00  -.2322699E+00   .1029947E+01   .1340198E-02   .1305186E+00
1700   1    .1005883E+00  -.5440810E-01  -.4188790E+00   .2253837E+00  -.1386364E+00
1800   2    .9977760E+00   .1029084E+01  -.8100021E-01   .5415959E-03  -.6003663E+00
1900   3   -.2271720E+00  -.2369821E+00   .1031042E+01   .2899263E-03   .1218179E+00
2000   1    .9729968E-01  -.6610241E-01  -.5235988E+00   .1966588E+00  -.1923716E+00
2100   2    .9979270E+00   .1025890E+01  -.1381673E+00   .5000537E-03  -.6002711E+00
2200   3   -.2092135E+00  -.2620306E+00   .1030629E+01  -.2110958E-02   .1102132E+00
2300   1    .9300203E-01  -.7623009E-01  -.6283185E+00   .1398962E+00  -.2356202E+00
2400   2    .1005483E+01   .102325[illegible]E+01  -.1975938E+00  -.1257294E-02  -.59987[illegible]1E+00
2500   3   -.3959048E+00  -.3209001E+00   .1028985E+01  -.5511206E-02   .1011199E+00
2600   1    .8731204E-01  -.8487507E-01  -.7330383E+00   .6810752E-01  -.2464628E+00
2700   2    .1017983E+01   .1021275E+01  -.2421879E+00  -.3708640E-02  -.5991074E+00
2800   3   -.6347227E+00  -.3778387E+00   .1026544E+01  -.9095525E-02   .9855563E-01
2900   1    .8008766E-01  -.9231333E-01  -.8377580E+00   .7477900E-02  -.2276362E+00
3000   2    .1029067E+01   .1020127E+01  -.2648933E+00  -.5146658E-02  -.5979988E+00
3100   3   -.8234540E+00  -.3865944E+00   .1023555E+01  -.1207957E-01   .1015647E+00
3200   1    .7144016E-01  -.9876676E-01  -.9424778E+00  -.3567305E-01  -.1962318E+00
3300   2    .1036653E+01   .1020007E+01  -.2720005E+00  -.5376520E-02  -.5966297E+00
3400   3   -.1024584E+01  -.3442213E+00   .1020069E+01  -.1407044E-01   .1065337E+00
3500   1    .6158783E-01  -.1042907E+00  -.1047198E+01  -.6534221E-01  -.1586447E+00
3600   2    .1041305E+01   .1020983E+01  -.2670684E+00  -.4768851E-02  -.5951080E+00
3700   3   -.1324703E+01  -.2781963E+00   .1016111E+01  -.1499069E-01   .1113376E+00
3800   1    .5071863E-01  -.1088297E+00  -.1151917E+01  -.8108[illegible]7E-01  -.1162588E+00
3900   2    .1042699E+01   .1022812E+01  -.2472945E+00  -.3561[illegible]E-02  -.5935497E+00
4000   3   -.1750850E+01  -.2162065E+00   .1011814E+01  -.1477011E-01   .1158369E+00
4100   1    .3896382E-01  -.1123276E+00  -.1256637E+01  -.8049241E-01  -.7350396E-01
4200   2    .1040407E+01   .1024973E+01  -.2091745E+00  -.2133289E-02  -.5920854E+00
4300   3   -.2265229E+01  -.1680927E+00   .1007493E+01  -.1319614E-01   .1203060E+00
4400   1    .2645479E-01  -.1147821E+00  -.1361357E+01  -.6403464E-01  -.3700986E-01
4500   2    .1035503E+01   .1026886E+01  -.1524454E+00  -.9640050E-03  -.5908674E+00
```

TABLE 10 - (continued)

4600	3	-.2774616E+01	-.1236639E+00	.1003665E+01	-.1006735E-01	.1243714E+00
4700	1	.1338044E-01	-.1162293E+00	-.1466077E+01	-.3535184E-01	-.1254040E-01
4800	2	.1030611E+01	.1028144E+01	-.8053472E-01	-.3055445E-03	-.5900534E+00
4900	3	-.3155131E+01	-.6822705E-01	.1000980E+01	-.5482609E-02	.1272438E+00
5000	1	-.3423824E-11	-.1167067E+00	-.1570796E+01	.9717779E-11	-.3919891E-02
5100	2	.1028565E+01	.1028572E+01	.2101379E-10	-.1119698E-03	-.5897664E+00
5200	3	-.3296611E+01	.1903118E-10	.1000008E+01	.1400675E-11	.1282839E+00
5300	2-0.540000					
5400	1	.1195187E+00	0.	0.	.1726273E+00	0.
5500	2	.1033713E+01	.1048012E+01	0.	-.5734943E-02	-.5400000E+00
5600	3	-.1740499E+01	0.	.1014791E+01	0.	.1470473E+00
5700	1	.1191601E+00	-.1535795E-01	-.1047198E+00	.1772038E+00	-.1206576E-01
5800	2	.1032026E+01	.1047081E+01	.1461010E-01	-.5599279E-02	-.5401326E+00
5900	3	-.1518471E+01	-.4281959E-01	.1015801E+01	.2532992E-02	.1459863E+00
6000	1	.1180539E+00	-.3042280E-01	-.2094395E+00	.1877014E+00	-.3271569E-01
6100	2	.1027007E+01	.1044418E+01	.1453239E-01	-.5038358E-02	-.5404757E+00
6200	3	-.9943291E+00	-.8043367E-01	.1018293E+01	.4022324E-02	.1420645E+00
6300	1	.1162987E+00	-.4481263E-01	-.3141593E+00	.1959246E+00	-.6581593E-01
6400	2	.1019780E+01	.1040483E+01	-.6936500E-02	-.3939546E-02	-.5408890E+00
6500	3	-.4560700E+00	-.1161479E+00	.1020887E+01	.3873435E-02	.1343139E+00
6600	1	.1138514E+00	-.5808049E-01	-.4188790E+00	.1928358E+00	-.1103753E+00
6700	2	.1012894E+01	.1035865E+01	-.4796494E-01	-.2676672E-02	-.5412007E+00
6800	3	-.1365952E+00	-.1631869E+00	.1022421E+01	.2080112E-02	.1231138E+00
6900	1	.1104692E+00	-.6985761E-01	-.5235988E+00	.1689584E+00	-.1599221E+00
7000	2	.1009655E+01	.1031139E+01	-.1013624E+00	-.2015065E-02	-.5412615E+00
7100	3	-.1269329E+00	-.2349296E+00	.1022553E+01	-.9208378E-03	.1113327E+00
7200	1	.1057922E+00	-.8000459E-01	-.6283185E+00	.1197554E+00	-.1984719E+00
7300	2	.1012270E+01	.1026784E+01	-.1553451E+00	-.2455138E-02	-.5409783E+00
7400	3	-.3184498E+00	-.3224631E+00	.1021645E+01	-.4491291E-02	.1033845E+00
7500	1	.9953330E-01	-.8862767E-01	-.7330383E+00	.5818337E-01	-.2093423E+00
7600	2	.1019080E+01	.1023169E+01	-.1970950E+00	-.3464412E-02	-.5403230E+00
7700	3	-.4661877E+00	-.3861478E+00	.1020180E+01	-.8029229E-02	.1019603E+00
7800	1	.9161058E-01	-.9594043E-01	-.8377580E+00	.3955355E-02	-.1950477E+00
7900	2	.1026277E+01	.1020569E+01	-.2229923E+00	-.4124788E-02	-.5393205E+00
8000	3	-.5293835E+00	-.3977186E+00	.1018335E+01	-.1112082E-01	.1058141E+00
8100	1	.8212461E-01	-.1020975E+00	-.9424778E+00	-.3674202E-01	-.1666526E+00
8200	2	.1032070E+01	.1019121E+01	-.2359362E+00	-.4142793E-02	-.5380306E+00
8300	3	-.6824672E+00	-.3690810E+00	.1015992E+01	-.1351804E-01	.1120865E+00
8400	1	.7123166E-01	-.1071360E+00	-.1047198E+01	-.6320235E-01	-.1296369E+00
8500	2	.1035474E+01	.1018731E+01	-.2349075E+00	-.3510063E-02	-.5365384E+00
8600	3	-.1017475E+01	-.3273791E+00	.1013031E+01	-.1498298E-01	.1193080E+00
8700	1	.5904541E-01	-.1110407E+00	-.1151917E+01	-.7316435E-01	-.8907492E-01
8800	2	.1035281E+01	.1019042E+01	-.2159651E+00	-.2411822E-02	-.5349592E+00
8900	3	-.1465833E+01	-.2884922E+00	.1009573E+01	-.1517921E-01	.1273743E+00
9000	1	.4564362E-01	-.1138421E+00	-.1256637E+01	-.6700109E-01	-.5243209E-01

3.4 CBLGEO

CBLGEO sets up the panels for the potential calculation. It computes the (S, x, y) coordinates of the panel corners and stores these values in FILE 7 TITLE = "CBL3DOPT".

Input data is again entered through the keyboard in response to the computer's prompting or it is entered from one of two stored files: FILE 8 TITLE = "BNSAL", the BNS file created by MAP 22, or FILE 10 TITLE = CBL3DINP", a file created by the present program during a previous run. This latter file has the same entries as TBL3DINP.

Table 11 shows the example problem being run from the terminal with entries made remotely NINPT = 2. Note that there is no need to enter START or SFINIS as these values are set by the program. After NTH = 10 is entered and echoed the ten values of θ are printed at the terminal.

There are two output files: FILE 11 TITLE = "CBLTAPE" and CBL3DOPT. CBLTAPE is created in the process of entering the data from the terminal; it is exactly CBL3DINP with a different name. CBL3DOPT contains all the panel data to be passed onto the potential program. Table 12 shows the output for the example case. Only the first four panels are shown.

Line 100: identifies the data

Line 200: N, KPK, BL

Line 300: START, SFINIS, NSTAT, NTH

Line 400: 1 first panel

Note that lines 400, 1000, 1600, 2200 give the panel number for the next five lines of data. Lines beginning with 1, 2, 3, 4, and 5 have the following nomenclature:

1	S(I)	x(I,J)	y(I,J)	TTH(J)
2	S(I+1)	x(I+1,J)	y(IH,J)	TTH(J)
3	S(I+1)	x(I+1, J+1)	y(I+1, J+1)	TTH(J+1)
4	S(I)	x(I, J+1)	y(I, J+1)	TTH(J+1)
5	SE(MD)	XE(MP)	YE(MP)	TTE(MP)

TABLE 11 - TERMINAL DISPLAY FOR CBL

```
       SSPA MODEL 720
  7   11   1.00000
 1.05407   0.00946  -0.06386  -0.17969  -2.17726   1.30080   1.75289
-2.35902   0.99482   1.35102  -1.27218
 0.09199  -0.01678   0.12962   0.23842  -1.63679  -0.65025   1.62619
 1.30540  -1.17223  -0.99895   0.67261
-0.14817  -0.01482   0.27346   0.19518   0.85676  -0.34804  -2.45327
 0.50034   2.46250  -0.32998  -0.98946
-0.00560  -0.00706   0.00830   0.22146   0.35303  -0.73910  -1.28967
 1.22848   1.99487  -0.70358  -1.06058
 0.00609  -0.01300  -0.15698   0.25859   1.06743  -1.09828  -2.79267
 1.64747   3.20478  -0.79404  -1.32815
 0.00030  -0.01009  -0.05990   0.14915   0.62418  -0.50594  -1.88688
 0.62018   2.24264  -0.25234  -0.92001
 0.00096   0.00142  -0.01278  -0.01756   0.05647   0.09382  -0.03964
-0.17823  -0.08303   0.10064   0.07818
NINPT DENOTES THE TYPE OF INPUT
NINPT=1  CARD DATA
      2  REMOTE
      3  STORED DATA


NINPT     1I1
#?
2
DO YOU WISH TO ENTER NEW BL?
IF YES TYPE 1 NO TYPE 0
1
BL=    1F10.5
 0.1415
  0.14150
NSTAT=       1I3
 25
  25
NTH=         1I3
 10
  10
  0.00000  -0.17453  -0.34907  -0.52360  -0.69813  -0.87266  -1.04720
 -1.22173  -1.39626  -1.57080
```

TABLE 12 - FILE CBL3DOPT

```
 100              SCPA MODEL 7 [illegible]
 200               [illegible]        11    [illegible]
 300       -1.00000    1.00000              [illegible]          [illegible]
 400         1
 500         1   -1.00000     0.00000     0.00000     0.00000
 600         2   -0.91667     0.02069     0.00000     0.00000
 700         3   -0.91667     0.01938    -0.01541    -0.17453
 800         4   -1.00000     0.00000    -0.06832    -0.17453
 900         5   -0.95833     0.01075    -0.06611    -0.08727
1000         2
1100         1   -1.00000     0.00000    -0.06832    -0.17453
1200         2   -0.91667     0.01938    -0.01541    -0.17453
1300         3   -0.91667     0.01662    -0.03242    -0.34907
1400         4   -1.00000     0.00000    -0.01680    -0.34907
1500         5   -0.95833     0.00279    -0.04932    -0.26180
1600         3
1700         1   -1.00000     0.00000    -0.01680    -0.34907
1800         2   -0.91667     0.01662    -0.03242    -0.34907
1900         3   -0.91667     0.01474    -0.04934    -0.52360
2000         4   -1.00000     0.00000    -0.02352    -0.52360
2100         5   -0.95833     0.00710    -0.03357    -0.43633
2200         4
2300         1   -1.00000     0.00000    -0.02352    -0.52360
2400         2   -0.91667     0.01474    -0.04934    -0.52360
2500         3   -0.91667     0.01301    -0.06395    -0.69813
2600         4   -1.00000     0.00000    -0.03030    -0.69813
2700         5   -0.95833     0.00641    -0.04668    -0.61087
```

The lines 1 thru 4 give the corner coordinates [illegible]
fifth line gives the panel midpoint coordinates [illegible]
values of the potential will be given.

3.5 DOUBDD

DOUBDD computes the potential [illegible]
panel geometry from CBICFO. Since it requires [illegible]
actively simply set procedual time at [illegible]
ques with the appropriate time requirement. [illegible]
FILE 7 TITLE = "CBL3DOPT" which [illegible]
put file, FILE 8 TITLE = DOUBDOPT". [illegible]
appear for the example case, only the first [illegible]
follows:

Line 100: NP the total number of panels used

Line 200: NSMO number of panels [illegible]
panels in girth direction.

Line 300: x-coordinate at which the potential [illegible]
coordinate and PHI value of potential at [illegible]

typical of the remaining lines in the file.

3.6 TBLIPC

This program is mainly a utility program used to format [illegible]
and the geometry data for input into the boundary-layer [illegible]
is reasonably fast taking about a minute for slender body [illegible]
four minutes for the data from DOUBDD. There are two [illegible]
"TBL3DOPT" and FILE 12 TITLE = "DOUBDOPT". Both files are required, [illegible]
potential based on the doublet distribution is used, but [illegible]
the case of the slender body potential flow.

In the interactive operation there is only one data value [illegible]
the operator. If the data is from DOUBDD, the data is simply [illegible]
0. The number that appears on the screen identifies the data being formatted.

Table 14 shows the screen for the test case. The data is identified [illegible]
for the SSPA Model 720. In the next line N=7, FPK=11, and the beam to length

100	216			
200	24			
300	-.95414E+00	.11081E-01	-.61080E-02	-.98096E+00
400	-.95311E+00	.10129E-01	-.18814E-01	-.98506E+00
500	-.95275E+00	.88904E-02	-.31775E-01	-.98556E+00
600	-.99315E+00	.78013E-02	-.43626E-01	-.98103E+00
700	-.95303E+00	.65537E-02	-.54083E-01	-.98019E+00
800	-.95219E+00	.50183E-02	-.63442E-01	-.97167E+00
900	-.95182E+00	.35301E-02	-.70900E-01	-.96938E+00
1000	-.95195E+00	.22263E-02	-.75735E-01	-.95868E+00
1100	-.94743E+00	.87134E-03	-.81059E-01	-.99335E+00
1200	-.87387E+00	.31066E-01	-.84092E-02	-.92329E+00
1300	-.87413E+00	.28746E-01	-.25893E-01	-.91733E+00
1400	-.87427E+00	.25951E-01	-.43865E-01	-.91827E+00
1500	-.87415E+00	.23302E-01	-.60468E-01	-.92244E+00
1600	-.87396E+00	.19858E-01	-.74786E-01	-.92817E+00
1700	-.87385E+00	.15450E-01	-.87115E-01	-.92725E+00
1800	-.87309E+00	.10947E-01	-.97344E-01	-.91640E+00
1900	-.87128E+00	.66362E-02	-.10474E+00	-.93054E+00
2000	-.87031E+00	.22285E-02	-.10854E+00	-.92566E+00

TABLE 14 - TERMINAL DISPLAY FOR TBLIPC

```
SSPA MODEL 720
   7        11    0.14150
-0.60000   0.60000       21        16
16  -.1047198E+00  -.9549294E+01
IF THREE DIMENSIONAL POTENTIAL IS FROM MING'S PROGRAM
ENTER 1 SLENDER BODY ENTER 0
8?
1
    216
     24          9
    216
```

ratio is 0.14150. The initial station for the data is S = -0.6, and the last station is S=0.6. There are 21 stations to be used in the lengthwise direction, and 16 values of θ. The 16 values of θ are equally spaced with $\Delta\theta$ = -0.1047198 and $1/\Delta\theta$ = -9.549294. In the case shown the doublet distribution potential is used so a 1 appears as the answer to the question. There were 216 panels used in the doublet method: 24 in the lengthwise direction and 9 in the girth direction. The final 216 indicates that the program is reading the data for the 216 panels.

There is only one output file, FILE 7 TITLE = "TBLIPCOPT". A representative portion of it is shown in Table 15.

Line 100: Data indentification

Line 200: N, KPK, and BL

Line 300: START, SFINIS, NSTAT, and NTH

Line 400: The station number and the value of S at this station.

Lines beginning with 1, 2, and 3 have the data organized as follows:

1	x	y	θ	u_ϕ	u_θ
2	S	$du_\phi/d\xi_\phi$	$du_\phi/d\xi_\theta$	$du_\theta/d\xi_\phi$	$du_\theta/d\xi_\theta$
3	K_ϕ	K_θ	H	F	G

3.7 TBLSOL

This final program of the suite, TBLSOL, computes the boundary-layer momentum thickness, TH11, the tangent of the crossflow angle T, and the shape parameter H. Input data is entered either through the keyboard in answer to prompting from the computer, or it is read from the stored file: FILE 8 TITLE ="TBLIPCOPT". This file is created by TBLIPC and is described in Section 3.6.

TABLE 15 - FILE TBLIPCOPT

```
 100          SSPA MODEL 720
 200             7           11     0.14159
 300     -0.60000     0.60000             21             16
 400     1-(.600000)
 500  1   .1076487E+00   0.              0.              .1039358E+01   0.
 600  2  -.6000000E+00    .1421311E+00    .3599764E-01   0.             -.1422674E+00
 700  3  -.1746801E+01   0.               .1024530E+01   0.              .1319586E+00
 800  1   .1070765E+00  -.1303975E-01   -.1047198E+00    .1038446E+01   .1965947E-02
 900  2  -.6000375E+00    .1419029E+00    .1218052E+00    .2077872E+00   .3065287E+00
1000  3  -.1503572E+01  -.1295564E+00    .1025467E+01    .7161124E-03   .1329239E+00
1100  1   .1055012E+00  -.2772637E-01   -.2094395E+00    .1035957E+01  -.8533513E-02
1200  2  -.6001429E+00    .1390912E+00    .2094959E+00    .3796330E+00   .1372600E+01
1300  3  -.9776511E+00  -.2079391E+00    .1027713E+01    .1298230E-02   .1337157E+00
1400  1   .1032734E+00  -.4142935E-01   -.3141593E+00    .1032607E+01  -.3647421E-01
1500  2  -.6002810E+00    .1339973E+00    .2517147E+00    .4871595E+00   .2650982E+01
1600  3  -.4968123E+00  -.2322699E+00    .1029947E+01    .1340196E-02   .1305186E+00
1700  1   .1005482E+00  -.5445510E-01   -.4188790E+00    .1029034E+01  -.8100021E-01
1800  2  -.6003663E+00    .1120477E+00    .2632725E+00    .5461084E+00   .3935456E+01
1900  3  -.2271720E+00  -.2354821E+00    .1031042E+01    .2899263E-03   .1219179E+00
2000  1   .9729366E-01  -.5610241E-01   -.5235988E+00    .1025890E+01  -.1381673E+00
2100  2  -.6002711E+00    .6639975E-01    .3527400E+00    .6099541E+00   .5051057E+01
2200  3  -.2092135E+00  -.2620306E+00    .1030629E+01   -.2110958E-02   .1102132E+00
2300  1   .9300203E-01  -.7623009E-01   -.6283185E+00    .1023250E+01  -.1975938E+00
2400  2  -.5998723E+00    .3373500E-01    .7179096E+00    .7021724E+00   .4911611E+01
2500  3  -.3839948E+00  -.3200001E+00    .1028985E+01   -.5511206E-02   .1011199E+00
2600  1   .8731294E-01  -.9487507E-01   -.7330383E+00    .1021275E+01  -.2421879E+00
2700  2  -.5971074E+00    .3159211E-01    .1512975E+00    .7519159E+00   .3250406E+01
2800  3  -.6347227E+00  -.3777377E+00    .1026544E+01   -.9095525E-02   .9555567E-01
2900  1   .8006786E-01  -.9031733E-01   -.8377580E+00    .1020127E+01  -.2648933E+00
3000  2  -.5279980E+00    .7386974E-02    .5960981E-01    .7012231E+00   .1401517E+01
3100  3  -.8234340E+00  -.3363944E+00    .1023555E+01   -.1207957E-01   .1015647E+00
3200  1   .7144016E-01  -.9776676E-01   -.9424778E+00    .1020007E+01  -.2720005E+00
3300  2  -.5765397E+00  -.1480170E-01   -.3836436E-01    .6057540E+00   .9748403E-01
3400  3  -.1024584E+01  -.3442213E+00    .1020069E+01   -.1407044E-01   .1055337E+00
```

Table 16 shows the example problem being run from the terminal. The first two rows of numbers are the polynomial coefficients in the skin friction formula given by Head and Patel[9]. These are fixed values and appear as data items in the program itself. In the example case the number of stations at which the boundary layer is to be calculated is NS=21 with only one iteration at each step, NT=1. TOL is the present value of tolerance used in the program; any number less than TOL is considered zero. There should be no need to change the value from 10^{-9}. If a larger TOL is needed, it is likely that something is going wrong in the computation or the initial data is not physically correct.

In the example, constant values were used for the initial data; these were TH11 = 0.0004, T=0, and H=1.4 where TH11 has been nondimensionalized with respect to the semi-hull length. These values are entered at the appropriate prompting.

The initial data will generally be obtained from experimental data. In the four cases that this program was used with experimental data, different interpolation methods were used. This amounted to writing a subroutine to interpolate the experimental data in each case as a function of θ and simply calling the subroutine to set the initial values. Enter a call for this subroutine at line 2500 in Subroutine INITIAL and change 2500 from STOP to RETURN. If initial data is entered as a function of θ merely respond with a 1 and the program will ask for the initial data at $j\Delta\theta$, $j = 0, 1, 2, \ldots,$ NTH.

The output for the test case is presented in Table 17.

Line 100: NS and NI

Line 200: Title

Line 300: N, KPK, and BL

Line 400: START, SFINIS, NSTAT, and NTH

Line 500: TH11, T, and H

Line 1200: Station 1, S at initial station

Lines 1300 Through 6100 have the following format:

1	x	y	θ	u_ϕ	u_θ
2	s	$du_\phi/d\ell_\phi$	$du_\theta/d\ell_\theta$	$du_\theta/d\ell_\phi$	$du_\phi/d\ell_\theta$
3	K	K	H	F	G

Lines 6200 through 7900 are typical of the output for each station. Line 6200 has

TABLE 16 - TERMINAL DISPLAY FOR TBLSOL

```
RUN TBLSOL
#RUNNING 4359
-0.000701  0.028345 -0.386768  0.019521
-0.001953  0.062588 -0.834800  0.191511
NS IS NUMBER OF STATIONS AT WHICH BOUNDARY LAYER IS
TO BE CALCULATED
NS=   1I3
#?
 21
21
NI IS NUMBER OF ITTERATIONS AT EACH STEP
NI=     1I3
  1
 1
TOL=    .10000E-03
IF THE INITIAL VALUES ARE CONSTANT ENTER 0
OTHERWISE ENTER A 1
0
TH11=     1F10.5
 .0004
  0.00040
T= TAN(BETA)=      1F10.5
 0.0
  0.00000
H=        1F10.5
```

TABLE 17 - PRINTER FILE TBLSOL

```
100    21  10
200          SSPA MODEL 720
300            7          11   0.14150
400    -0.60000   0.60000         21          16
500     0.00040   0.00000    1.40000
600  1INITIAL VALUES
700
800   TH11=    0.0004      T=    0.0000      H=    1.4000
900
1000   1-0.600000
1100   1    .1076487E+00  0.              0.               .1039358E+01  0.
1200   2  -.6000000E+00   .1421311E+00   .6599764E-01  0.               -.1422674E+00
1300   3  -.1746801E+01  0.               .1024530E+01  0.                .1319586E+00
1400   1    .1070765E+00  -.1383975E-01  -.1047198E+00   .1038446E+01   .1965947E-02
1500   2  -.6000375E+00   .1419029E+00   .1218052E+00   .2077872E+00   .3065287E+00
1600   3  -.1503672E+01  -.1295564E+00   .1025467E+01   .7161124E-03   .1329239E+00
1700   1    .1055012E+00  -.2772837E-01  -.2094395E+00   .1035967E+01  -.8533613E-02
1800   2  -.6001429E+00   .1390912E+00   .2084958E+00   .3796330E+00   .1372600E+01
1900   3  -.9778611E+00  -.2079391E+00   .1027713E+01   .1298230E-02   .1337157E+00
2000   1    .1032734E+00  -.4142935E-01  -.3141593E+00   .1032607E+01  -.3647421E-01
2100   2  -.6002810E+00   .1298973E+00   .2517947E+00   .4871595E+00   .2650982E+01
2200   3  -.4966123E+00  -.2322699E+00   .1029947E+01   .1340198E-02   .1305186E+00
2300   2-0.540000
2400 2      THATA        X          Y          S           TH11              T                  H
6200      0.0000     0.1195     0.0000    -0.5400     .56653797E-03    0.                  .15446615E+01
6300     -0.1047     0.1192    -0.0154    -0.5401     .56673740E-03     .11607606E-01      .15327303E+01
6400     -0.2094     0.1181    -0.0304    -0.5405     .56799616E-03     .10985056E-01      .15079857E+01
6500     -0.3142     0.1163    -0.0448    -0.5409     .56935764E-03    -.42664793E-02      .14840518E+01
6600     -0.4189     0.1139    -0.0581    -0.5412     .56752441E-03    -.31504725E-01      .14629403E+01
6700     -0.5236     0.1105    -0.0699    -0.5413     .56117968E-03    -.67427577E-01      .14446639E+01
6800     -0.6283     0.1058    -0.0800    -0.5410     .55403075E-03    -.10651129E+00      .14341994E+01
6900     -0.7330     0.0995    -0.0886    -0.5403     .55261012E-03    -.14042963E+00      .14376116E+01
7000     -0.8378     0.0916    -0.0959    -0.5393     .55793645E-03    -.16423704E+00      .14528439E+01
7100     -0.9425     0.0821    -0.1021    -0.5380     .56499103E-03    -.17867669E+00      .14724043E+01
7200     -1.0472     0.0712    -0.1071    -0.5365     .57059685E-03    -.18476568E+00      .14936899E+01
7300     -1.1519     0.0590    -0.1110    -0.5350     .57523754E-03    -.18079522E+00      .15193881E+01
7400     -1.2566     0.0456    -0.1138    -0.5334     .58074369E-03    -.16254177E+00      .15528803E+01
7500     -1.3614     0.0311    -0.1157    -0.5322     .58815735E-03    -.12539799E+00      .15928778E+01
7600     -1.4661     0.0158    -0.1167    -0.5313     .59551192E-03    -.68898905E-01      .16282236E+01
7700     -1.5708    -0.0000    -0.1170    -0.5310     .59867344E-03    0.                  .16426010E+01
```

the station number and station s coordinate. The remaining lines give the boundary -layer parameters around the girth at station s as a function of θ.

Running time for this program has always been less than five minutes.

4. CONCLUDING REMARKS

This report presents a suite of computer programs for computing three dimensional boundary layers for ships. Most of the technical details for carrying out the computations are explained and demonstrated in the form of a worked out example. The computer programs are in a structured form that permits modification to empirical formulas as better empirical approximations are developed, e.g. an improved crossflow model when new experimental data becomes available. Alternatively, individual programs described in this report, such as MAP22, and DOUBDD can be used independently or with other methods for calculating the boundary layer.

There are some choices that must be made in the input data to these programs. The number of hull offsets to be used in obtaining the rational hull approximation together with the degree of the lengthwise approximating polynomial and the degree of the conformal mapping are the most difficult choices to be made. The values used in the example are a good starting guess; however, these numbers should be determined by comparing the agreement of the approximating offsets with the physical hull offsets and by examining the smoothness of the approximating hull shape. Changing the degree of the lengthwise polynomial and the number of parameters in the conformal mapping will change the smoothness of the approximating hull form. 250 panels should be used in the potential calculation; the panel dimensions should be chosen so that the longest panel dimension is in the direction of the freestream flow. In the boundary layer calculation 40 girthwise values should be used.

von Kerczek and Langan concluded that the present boundary layer calculations method can predict boundary layers on relatively fine double ship models with fair accuracy to within a distance of the stern of about 10 percent of the ship's length for the SSPA Model 720. In the stern area, the boundary layer thickens very rapidly and approaches separation. Calculation of this near-separated boundary layer region must await further developments of boundary layer theory.

REFERENCES

1. Tuck, E.O. and C. von Kerczek, "Streamlines and Pressure Distribution on Arbitrary Ship Hulls at Zero Froude Number," Journal of Ship Research, Vol. 12, pp. 231-236 (1968).

2. von Kerczek, C., "Calculation of the Turbulent Boundary Layer on a Ship Hull," Journal of Ship Research, Vol. 17, pp. 106-120 (1973).

3. von Kerczek, C. and E.O. Tuck, "The Representation of Ship Hulls by Conformal Mapping Functions," Journal of Ship Research, Vol. 13, pp. 284-298 (1969).

4. Chang, M.S. and P.C. Pien, "Hydrodynamic Forces on a Body Moving Beneath a Free Surface," Proceedings of the First International Conference on Numerical Ship Hydrodynamics, J. Schot and N. Salvesen, editors, Gaithersburg (1975).

5. von Kerczek, C. and T.J. Langan, "An Integral Prediction Method for Three-Dimensional Turbulent Boundary Layers on Ships," David W. Taylor Naval Ship R&D Center Report 79/006 (1979).

6. Larsson, L., "Boundary Layers on Ships, Part IV: Calculations of the Turbulent Boundary Layer on a Ship Model," The Swedish State Shipbuilding Experimental Tank, Goteborg, Sweden, Report 47 (1974).

7. Cumpsty, N.A. and M.R. Head, "The Calculation of Three-Dimensional Turbulent Boundary Layers, Part I: Flow Over the Rear of an Infinite Swept Wing," The Aeronautical Quarterly, Vol. 18, pp. 55-84 (1965).

8. Smith, P.D., "An Integral Prediction Method for Three-Dimensional Compressible Turbulent Boundary Layers," Royal Aeronautical Establishment, Reports and Memoranda 3739 (1972).

9. Head, M.R. and V.C. Patel, "Improved Entrainment Method for Calculating Turbulent Boundary Layer Development," Aeronautical Research Council (Great Britain), R and M 3643 (1968).

APPENDIX

PROGRAM LISTINGS

```
FILE  5(TITLE="PFN",KIND=DISK,FILETYPE=7)
FILE  8(TITLE="ANS",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
FILE  9(TITLE="BNS",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
FILE  6(KIND=PRINTER,MAXRECSIZE=22)
 $RESET FREE
C      PROGRAM MAP22
         COMMON /A1/ XMS(50,50),YMS(50,50)
         COMMON /A2/ AN(50,10)
         COMMON /A3/ BN(25,10)
         COMMON /A4/ BS(25,25)
C
C   MAP FITS AN ANALYTIC SURFACE TO A SHIP HULL
C     POINTS AT THE WATERLINE AND AT THE KEEL CONSTRAINED - ALL STATIONS

C         XMS       X OFFSET OF A POINT ON THE HULL
C         YMS       Y OFFSET OF A POINT ON THE HULL
C         AN(I,J)    MAPPING COEFFICIENTS AT STATION I
C         BN(I,J)   CALCULATED A SUB MN
C         BS(I,J)   COEFFICIENT MATRIX
C         KT(I)     NUMBER OF OFFSETS GIVEN AT STATION I
      COMMON /A5/ KT(50)
      COMMON /A6/ IFS(50)
      COMMON /A7/ JFS(50)
      COMMON /A8/ TITLE(12)
      COMMON /A9/ THETA(50)
      COMMON /B1/ S(50)
      COMMON /B2/ AP(10)
      COMMON /B3/ B(50),H(50),SA(50)
      CALL INPUT(K,N,IST1,IST2,KP1,KP2,TOL,BL)
      CALL CONTRA(K,N,TOL)
      IF(K.LE.2) GO TO 5
      CALL POLY(1,IST1,N,KP1)
    5 CALL OUTPUT(1,IST1,N,KP1,BL)
      IF(IST1-K) 1,2,2
    1 CALL POLY(IST2,K,N,KP2)
      CALL OUTPUT(IST2,K,N,KP2,BL)
      IF(IST2-IST1) 2,2,3
    3 DO 4 J=1,N
        BN(1,J)=BL*AN(IST1,J)
    4 BN(2,J)=0.0
      CALL OUTPUT(IST1,IST2,N,2,BL)
    2 STOP
      END
```

```
      SUBROUTINE INPUT(K,N,IST1,IST2,KP1,KP2,TOL,BL)

C   SUBROUTINE INPUT CONTAINS ALL READ STATEMENTS FOR MAP

C        XMS        X OFFSET OF A POINT ON THE HULL
C        YMS        Y OFFSET OF A POINT ON THE HULL
C        AN(I,J)     MAPPING COEFFICIENTS AT STATION I
C        BN(I,J)    CALCULATED A SUB MN
C        BS(I,J)    COEFFICIENT MATRIX
C        KT(I)      NUMBER OF OFFSETS GIVEN AT STATION I

C        IFS(I)     1 NO CONSTRANTS AT STATION I
C                   2 SURFACE AT STATION I TO PASS THROUGH OFFSETS
C                   3 AT STATION I SLOPE WITH RESPECT TO S IS 0
C                   4 2 AND 3 COMBINED
C                   5 AT STATION I SLOPE AND CURVATURE ARE 0
C                   6 2,3,AND 5 COMBINED

C        JFS(I)     0 AN READ IN AT STATION I
C                   1 COMPUTE AN

C        TITLE      IDENTIFIES COMPUTATIONS- 144 CHARACTERS MAX
C        THETA(J)   PHI SUB J AT A GIVEN STATION
C        S(I)       S AT STATION I
C        AP         MAPPING COEFFICIENTS AT A GIVEN STATION
C        B(I)       B EAM AT STATION I
C        H(I)       DRAFT AT STATION I
C        SA(I)      CROSS SECTIONAL AREA AT STATION I

      COMMON /A1/ XMS(50,50),YMS(50,50)
      COMMON /A2/ AN(50,10)
      COMMON /A5/ KT(50)
      COMMON /A6/ IFS(50)
      COMMON /A7/ JFS(50)
      COMMON /A8/ TITLE(12)
      COMMON /B1/ S(50)
      DIMENSION IX(50),IY(50)
      READ(5,24) (TITLE(J),J=1,12)
      READ(5,3) IST1,IST2,KP1,KP2
      READ(5,1) K,N,TOL,BL
      DO 58 I=1,K
      DO 58 J=1,N
   58 AN(I,J)=0.0
      READ(5,3) (IFS(I),I=1,K)
      READ(5,3) (JFS(I),I=1,K)
```

```
      READ(5,25)(KT(I),I=1,K)
      DO 33 I=1,K
      M=KT(I)
      READ(5,4) (IX(J),IY(J),J=1,M)
      DO 34 J=1,M
      XMS(I,J)=FLOAT(IX(J))
   34 YMS(I,J)=FLOAT(IY(J))
   33 CONTINUE
      XMAX=XMS(10,1)
      DO 35 I=1,14
      M=KT(I)
      DO 35 J=1,M
      XMS(I,J)=XMS(I,J)/XMAX
   35 YMS(I,J)=YMS(I,J)/XMAX
      DO 36 I=15,K
      M=KT(I)
      DO 36 J=1,M
      XMS(I,J)=-XMS(I,J)/XMAX
   36 YMS(I,J)=YMS(I,J)/XMAX
      READ(5,2) (S(I),I=1,K)
      DO 31 I=1,K
      IF(JFS(I)) 31,32,31
   32 READ(5,2) (AN(I,J),J=1,N)
   31 CONTINUE
      WRITE(6,40)
      WRITE(6,24) (TITLE(J),J=1,12)
      WRITE(9,24) (TITLE(J),J=1,12)
      WRITE(6,19)
      DO 21 I=1,K
      M=KT(I)
      WRITE(6,22) I,S(I)
   21 WRITE(6,23)(XMS(I,J),YMS(I,J),J=1,M)
  100 RETURN
    1 FORMAT(2I10,2F10.5)
    4 FORMAT(6(1X,2I6))
    2 FORMAT (7F10.5)
3     FORMAT(14I5)
   25 FORMAT(7I10)
   24 FORMAT(12A6)
   40 FORMAT(1H1)
   19 FORMAT(19H  THE INPUT OFFSETS    ///)
   22 FORMAT(//56H                    XMS                YMS
     1S(,I2,2H)=F5.2//)
   23 FORMAT(F30.5,F16.5)
      END
```

```
      SUBROUTINE CONTRA(K,N,TOL)

C   SUBROUTINE CONTRA COMPUTES THE PARAMETERS IN THE CONFORMAL MAP OF TH
C   UNIT CIRCLE ONTO THE CROSS SECTION

C        XMS       X OFFSET OF A POINT ON THE HULL
C        YMS       Y OFFSET OF A POINT ON THE HULL
C        AN(I,J)    MAPPING COEFFICIENTS AT STATION I
C        BS(I,J)   COEFFICIENT MATRIX
C        KT(I)     NUMBER OF OFFSETS GIVEN AT STATION I

C        JFS(I)    0 AN READ IN AT STATION I
C                  1 COMPUTE AN

C        THETA(J)  PHI SUB J AT A GIVEN STATION
C        AP        MAPPING COEFFICIENTS AT A GIVEN STATION
C        B(I)      B EAM AT STATION I
C        H(I)      DRAFT AT STATION I
C        SA(I)     CROSS SECTIONAL AREA AT STATION I

      COMMON /A1/ XMS(50,50),YMS(50,50)
      COMMON /A2/ AN(50,10)
      COMMON /A4/ BS(25,25)
       COMMON /A5/ KT(50)
       COMMON /A7/ JFS(50)
       COMMON/A9/ THETA(50)
       COMMON /B2/ AP(10)
      COMMON /B3/ B(50),H(50),SA(50)
      DIMENSION ANU(25,1),IP(25)
    5 FORMAT(23H THE MATRIX IS SINGULAR I5,F10.5)
   70 FORMAT(36H BEST FIT AT THIS NN NOT YET REACHED//)
  100 FORMAT(///13H  INITIAL AN(I2,4H,J)=,3F9.5,21H    A-EA COEFFICIENT=
     1F9.5//)
  104 FORMAT(10F12.8)
  107 FORMAT(18H NO CONVERGENCE AT I5)
  108 FORMAT(17H NO LEWIS FORM AT I5)
  110 FORMAT(6H  DUT=F8.5,5H  AN=11F9.5)
      THETA(1)=0.0
      PI=-3.14159265
      DO 103 L=1,K
      M=KT(L)
      B(L)=XMS(L,1)
      H(L)=-YMS(L,M)
      SUM=0.0
       DO 112 J=2,M
```

```
  112  SUM=SUM+0.5*(XMS(L,J)-XMS(L,J-1))*(YMS(L,J)-YMS(L,J))
       SA(L)=2.*SUM
  103 CONTINUE
      DO 54 I=1,K
      IF(JFS(I)) 11,54,11
   11 AN(I,2)=0.5*(B(I)-H(I))
      BH=B(I)+H(I)
      HBB=BH**2+8.*(2.*SA(I)/PI+B(I)*H(I))
      IF(HBB) 63,64,64
   63 WRITE(6,108) I
      AN(I,3)=0.0
      GO TO 54
   64 AN(I,3)=-0.25*(BH-SQRT(HBB))
   54  CONTINUE
       DO 10 I=1,K
      IF(JFS(I)) 12,13,12
   13 WRITE(6,6) I, (AN(I,J),J=1,N)
    6 FORMAT(//15H AN GIVEN AT I=I2,4H AN=10F9.5)
      GO TO 10
   12 M=KT(I)
      THETA(M)=-1.57079632
      MM=M-1
      INT=-1
      DUTP=0.0
      DO 91 J=1,10
   91 AP(J)=0.0
      FAC=B(I)*H(I)*2.
      IF(FAC) 86,92,86
   92 SIGMA=-1.0
      GO TO 65
   86 SIGMA=SA(I)/FAC
   65 WRITE(6,100) I,AN(I,1),AN(I,2),AN(I,3),SIGMA
      NN=2
   67 NN=NN+1
      NN2=NN-2
      IF(NN-N) 68,68,21
   68 KLK=0
   35 KLK=KLK+1
      IF(KLK-20) 55,55,56
   56 WRITE(6,107) I
      IF(NN-N) 67,10,10
   55 IF(INT) 1,1,2
    2 DO 28 J=1,NN2
      U=J+2
      UJ=(-1.)**( J+2)
```

```
      DO 29 JJ=1,NN2
      UN=JJ+2
      ON=(-1.)**(JJ+2)
      SUM=0.0
      DO 30 JJJ=2,MM
      TH1=THETA(JJJ)
      TH2=(3.-2.*U)*TH1
      TH3=(3.-2.*UN)*TH1
      C1=COS(TH1)
      C2=COS(TH2)
      C3=COS(TH3)
      S1=SIN(TH1)
      S2=SIN(TH2)
      S3=SIN(TH3)
   30 SUM=SUM + (C1*C1 - C1*C2 - C1*C3 + C2*C3 + ON*OJ*S1*S1 + ON*S1*S2
     1 + OJ*S1*S3 + S2*S3)
   29 BS(J,JJ)=SUM
   28 CONTINUE
      DO 26 J=1,NN2
      U=J+2
      OJ=(-1.)**(J+3)
      SUM=0.0
      DO 27 JJ=2,MM
      TH1=THETA(JJ)
      TH2=(3.-2.*U)*TH1
      C1=COS(TH1)
      C2=COS(TH2)
      S1=SIN(TH1)
      S2=SIN(TH2)
   27 SUM=SUM + (XMS(I,JJ) - B(I)*C1) * (C1-C2) +
     1 (YMS(I,JJ) - H(I)*S1) * (OJ*S1 - S2)
   26 ANU(J,1)=SUM
      CALL DECOMP(NN2,25,BS,IP)
      CALL SOLVE(NN2,25,BS,ANU,IP)
   52 DO 59 J=1,NN2
   59 AN(I,J+2)=ANU(J,1)
    1 SUM1=0.0
      SUM2=0.0
      DO 200 J=3,NN
      V1=1.-(-1.)**J
      V2=1.+(-1.)**J
      SUM1=SUM1+V1*AN(I,J)
  200 SUM2=SUM2+V2*AN(I,J)
      AN(I,1)=0.5*(B(I)+H(I)-SUM1)
      AN(I,2)=0.5*(B(I)-H(I)-SUM2)
```

```
      CALL ATHETA(I,NN,MM,IBTH,N)
      IF(IBTH) 21,21,4
    4 DUT=0.0
      DO 32 J=1,M
      SUM1=0.0
      SUM2=0.0
      DO 33 JJ=1,NN
      PJ=JJ
      ARG=(3.-2.*PJ)*THETA(J)
      SUM1=SUM1+AN(I,JJ)*COS(ARG)
   33 SUM2=SUM2+AN(I,JJ)*SIN(ARG)
   32 DUT=DUT+(XMS(I,J)-SUM1)**2+(YMS(I,J)-SUM2)**2
      TUD=DUT-DUTP
      IF(INT) 78,78,79
   78 INT=1
      WRITE(6,110) DUT,(AN(I,J),J=1,NN)
      DO 3 J=1,NN
    3 AP(J)=AN(I,J)
      DUTP=DUT
      GO TO 67
   79 IF(TUD) 88,88,87
   87 DO 89 J=1,NN
   89 AN(I,J)=AP(J)
      WRITE(6,90) DUT
   90 FORMAT(25H   FIT IS GOING BAD  DUT=F7.5)
      GO TO 67
   88 IF(ABS(TUD)-TOL) 69,69,66
   69 WRITE(6,110) DUT,(AN(I,J),J=1,NN)
      DO 77 J=1,NN
   77 AP(J)=AN(I,J)
      DUTP=DUT
      GO TO 67
   66 DUTP=DUT
       WRITE(6,110) DUT,(AN(I,J),J=1,NN)
      DO 76 J=1,NN
   76 AP(J)=AN(I,J)
      GO TO 35
   21 WRITE(8,111) (AN(I,J),J=1,N)
  111 FORMAT(7F10.5)
   10 CONTINUE
   50 RETURN
      END
#
```

```
      SUBROUTINE ATHETA(I,NN,MM,IBTH,N)
      COMMON /A1/ XMS(50,50),YMS(50,50)
      COMMON /A2/ AN(50,10)
      COMMON /A9/ THETA(50)
      IBTH=1
      M=MM+1
      KMM=1000
      OME1=0.0174533
      DO 11 J=2,MM
       THETA(J)=THETA(J-1)
      OMEG=THETA(J-1)
      DTHE=(OMEG-OME1)/10.
      JJJ=-1
      KKK=1
      LLL=1
   23 SUM1=0.0
      SUM2=0.0
      KKK=KKK+1
      DO 15 L=1,NN
      PN=L
      ARG=(3.-2.*PN)*THETA(J)
      SUM1=SUM1+AN(I,L)*COS(ARG)
   15 SUM2=SUM2+AN(I,L)*SIN(ARG)
      DU=(XMS(I,J)-SUM1)**2+(YMS(I,J)-SUM2)**2
      IF(JJJ) 12,13,14
   12 DU1=DU
      JJJ=0
      GO TO 16
   13 IF(DU-DU1) 17,18,18
   18 DU1=DU
      GO TO 16
   17 JJJ=1
      DU1=DU
      GO TO 16
   14 IF(DU-DU1) 20,19,19
   20 DU1=DU
   16 IF(KKK-KMM) 21,21,37
   21 THETA(J)=THETA(J)+DTHE
      GO TO 23
   37 IBTH=-1
      WRITE(6,4) I,J
    4 FORMAT(16H BAD THETA AT I=I2,3H J=I2//)
      GO TO 10
   19 IF(LLL-3) 24,25,25
   24 THETA(J)=THETA(J)-2.*DTHE
      DTHE=DTHE/10.
      JJJ=-1
      KKK=1
      LLL=LLL+1
      GO TO 23
   25 THETA(J)=THETA(J)-DTHE
      OME1=OMEG
   11 CONTINUE
   10 RETURN
      END
```

```
      SUBROUTINE POLY(KB,KE,N,KPK)
      COMMON /A2/ AN(50,10)
      COMMON /A3/ BN(25,10)
      COMMON /A4/ BS(25,25)
      COMMON /A6/ IFS(50)
      COMMON /B1/ S(50)
      COMMON /B3/ B(50),H(50),SA(50)
      DIMENSION IP(25),VE(25)
      INT=0
      DO 12 I=KB,KE
      IS=IFS(I)
      GO TO (12,13,13,28,28,45),IS
   28 INT=INT+2
      GO TO 12
   13 INT=INT+1
      GO TO 12
   45 INT=INT+3
   12 CONTINUE
      MP=KPK+INT
      KPK1=KPK+1
      WRITE(6,80) MP,KPK,INT
   80 FORMAT(//4H MP=I5,5H KPK=I5,5H INT=I5/)
      DO 27 I=1,MP
      DO 27 J=1,MP
   27 BS(I,J)=0.0
      DO 1 II=1,N
      DO 1 I=2,KPK
      JI=I-1
      SUM=0.0
      DO 2 J=KB,KE
      IS=IFS(J)
      GO TO (3,2,3,2,3,2),IS
    3 SUM=SUM+(S(J)**JI)*AN(J,II)
    2 CONTINUE
    1 BN(I,II)=SUM
      DO 7 II=1,N
      BN(1,II)=0.0
      DO 7 J=KB,KE
      IS=IFS(J)
      GO TO (5,7,5,7,5,7),IS
    5 BN(1,II)=BN(1,II)+AN(J,II)
    7 CONTINUE
      DO 4 II=1,N
      I=KPK1
      DO 4 J=KB,KE
```

```
      IS=IFS(J)
      GO TO (4,6,46,29,47,48),IS
   29 BN(I,II)=AN(J,II)
      BN(I+1,II)=0.0
      I=I+2
      GO TO 4
   46 BN(I,II)=0.0
      I=I+1
      GO TO 4
   47 BN(I,II)=0.0
      BN(I+1,II)=0.0
      I=I+2
      GO TO 4
   48 BN(I,II)=AN(J,II)
      I=I+1
      GO TO 47
    6 BN(I,II)=AN(J,II)
      I=I+1
    4 CONTINUE
      WRITE(6,71)
      DO 70 II=1,N
   70 WRITE(6,72) (BN(I,II),I=1,MP)
   71 FORMAT (//19H      THE VECTORS BN //)
   72 FORMAT (12F10.5)
      DO 8 I=1,KPK
      DO 8 J=1,KPK
      SUM=0.0
      IJ=I+J-2
      DO 9 M=KB,KE
      IS=IFS(M)
      GO TO (10,9,10,9,10,9),IS
   10 IF(IJ) 11,14,11
   14 IF(S(M)) 11,15,11
   15 SUM=SUM+1.0
      GO TO 9
   11 SUM=SUM+S(M)**IJ
    9 CONTINUE
    8 BS(I,J)=SUM
      L=KPK1
      DO 30 J=KB,KE
      IS=IFS(J)
      GO TO (30,31,49,32,50,51),IS
   31 BS(1,L)=1.0
      L=L+1
      GO TO 30
```

```
   49 BS(1,L)=0.0
      L=L+1
      GO TO 30
   50 BS(1,L)=0.0
      BS(1,L+1)=0.0
      L=L+2
      GO TO 30
   51 BS(1,L)=1.0
      L=L+1
      GO TO 50
   52 BS(1,L)=1.0
       BS(1,L+1)=0.0
      L=L+2
   30 CONTINUE
      DO 16 I=2,KPK
      L=KPK1
      I1=I-1
      DO 17 J=KB,KE
      IS=IFS(J)
      GO TO (17,18,52,33,55,59),IS
   18 BS(I,L)=S(J)**I1
      L=L+1
      GO TO 17
   52 I2=I1-1
      IF(I2) 53,54,53
   54 BS(I,L)=1.0
      L=L+1
      GO TO 17
   53 AI=I1
      BS(I,L)=AI*S(J)**I2
      L=L+1
      GO TO 17
   55 I2=I1-1
      I3=I1-2
      IF(I3) 56,57,58
   56 BS(I,L)=1.0
      BS(I,L+1)=0.0
      L=L+2
      GO TO 17
   57 AI1=I1
      BS(I,L)=AI1*S(J)
      BS(I,L+1)=2.0
      L=L+2
      GO TO 17
   58 AI1=I1
```

```
      AI2=I2
      BS(I,L)=AI1*S(J)**I2
      BS(I,L+1)=AI1*AI2*S(J)**I3
      L=L+2
      GO TO 17
   59 BS(I,L)=S(J)**I1
      L=L+1
      GO TO 55
   33 BS(I,L)=S(J)**I1
      I2=I1-1
      IF(I2) 34,35,34
   35 BS(I,L+1)=1.0
      GO TO 36
   34 AI=I1
      BS(I,L+1)=AI*S(J)**I2
   36 L=L+2
   17 CONTINUE
   16 CONTINUE
      L=KPK1
      DO 19 J=KB,KE
      IS=IFS(J)
      GO TO (19,20,60,37,62,65),IS
   20 DO 21 I=2,KPK
      I1=I-1
   21 BS(L,I)=S(J)**I1
      L=L+1
      GO TO 19
   60 BS(L,2)=1.0
      DO 61 I=3,KPK
      AI=I-1
      I2=I-2
   61 BS(L,I)=AI*S(J)**I2
      L=L+1
      GO TO 19
   62 BS(L,2)=1.0
      BS(L+1,2)=0.0
      DO 63 I=3,KPK
      AI=I-1
      I2=I-2
   63 BS(L,I)=AI*S(J)**I2
      L1=L+1
      BS(L1,3)=2.0
      DO 64 I=4,KPK
      AI1=I-1
      AI2=I-2
```

```
      I3=I-3
   64 BS(L1,I)=AI1*AI2*S(J)**I3
      L=L+2
      GO TO 19
   65 DO 66 I=2,KPK
      I1=I-1
   66 BS(L,I)=S(J)**I1
      L=L+1
      GO TO 62
   37 DO 44 I=2,KPK
      I1=I-1
      BS(L,I)=S(J)**I1
      I2=I1-1
      IF(I2) 38,39,38
   39 BS(L+1,I)=1.0
      GO TO 44
   38 AI=I1
      BS(L+1,I)=AI*S(J)**I2
   44 CONTINUE
   40 L=L+2
   19 CONTINUE
      L=KPK1
      DO 41 J=KB,KE
      IS=IFS(J)
      GO TO (41,42,67,43,68,69),IS
   43 BS(L,1)=1.0
      BS(L+1,1)=0.0
      L=L+2
      GO TO 41
   67 BS(L,1)=0.0
      L=L+1
      GO TO 41
   68 BS(L,1)=0.0
      BS(L+1,1)=0.0
      L=L+2
      GO TO 41
   69 BS(L,1)=1.0
      L=L+1
      GO TO 68
   42 BS(L,1)=1.0
      L=L+1
   41 CONTINUE
      WRITE(6,73)
   73 FORMAT (//18H     THE MATRIX BS //)
        DO 74 J=1,MP
```

```
   74 WRITE (6,72) (BS(J,I),I=1,MP)
      CALL DECOMP(MP,25,BS,IP)
      DO 23 I=1,N
      DO 24 J=1,MP
      VE(J)=BN(J,I)
   24 CONTINUE
      CALL SOLVE(MP,25,BS,VE,IP)
      DO 25 J=1,MP
      BN(J,I)=VE(J)
   25 CONTINUE
   23 CONTINUE
   26 RETURN
      END
```

```
      SUBROUTINE OUTPUT(KB,KE,N,KPK,BL)
      COMMON /A1/ XMS(50,50),YMS(50,50)
      COMMON /A2/ AN(50,10)
      COMMON /A3/ BN(25,10)
      COMMON /A5/ KT(50)
      COMMON /A8/ TITLE(12)
      COMMON /B1/ S(50)
      COMMON /B3/ B(50),H(50),SA(50)
      IF(KPK.LE.1) GO TO 13
      WRITE(6,70)
   70 FORMAT(//30H   THE POLYNOMIAL COEFFICIENTS//)
      DO 71 J=1,N
      IF(KPK-10) 72,72,73
   72 WRITE(6,74) J, (BN(I,J),I=1,KPK)
      GO TO 71
   73 WRITE(6,74) J, (BN(I,J),I=1,10)
      WRITE(6,75) (BN(I,J),I=11,KPK)
   71 CONTINUE
   74 FORMAT (4H J =, I2, 10F10.5)
   75 FORMAT(10H          ,10F10.5)
  111 FORMAT(7F10.5)
      DO 80 I=1,KE
   80 WRITE(8,111) (AN(I,J),J=1,N)
   13 DO 1 I=KB,KE
      IF(KPK.LE.1) GO TO 11
      ST=S(I)
      WRITE(6,7) I,ST
    7 FORMAT(//26H            X            Y,12H          S(I2,2H)=
     1F6.4/)
      DO 2 J=1,N
      SUM=0.0
      DO 3 II=2,KPK
      I1=II-1
    3 SUM=SUM+BN(II,J)*ST**I1
    2 AN(I,J)=SUM+BN(1,J)
   11 TH=0.03491
      DO 4 L=1,46
      X=0.0
      Y=0.0
      TH=TH-0.03491
      DO 5 J=1,N
      AJ=3-2*J
      X=X+AN(I,J)*COS(AJ*TH)
    5 Y=Y+AN(I,J)*SIN(AJ*TH)
    4 WRITE(6,6) X,Y
```

```
    6 FORMAT(F20.5,F10.5)
    1 CONTINUE
      IF(KPK.GT.1) GO TO 12
      RETURN
   12 WRITE(6,8)
    8 FORMAT(///17H           AN(I,J)//)
      DO 9 I=KB,KE
    9 WRITE(6,10) (AN(I,J),J=1,N)
   10 FORMAT(F20.5,9F10.5)
      WRITE(9,400) N,KPK,BL
      DO 76 J=1,N
   76 WRITE(9,111) (BN(I,J),I=1,KPK)
      RETURN
  400 FORMAT(2I5,1F10.5)
      END
```

```
      SUBROUTINE DECOMP(N,NDIM,A,IP)

C   SUBROUTINE DECOMP GAUSSIAN REDUCTION OF A TO TRIANGULAR FORM

C        N          NUMBER OF NON ZERO ROWS IN A
C        NDIM       DIMENSION OF A
C        A          MATRIX TO BE REDUCED
C        IP         NUMBER OF ROW BEING PIVIOTED WITH K ROW

      DIMENSION A(NDIM,NDIM),IP(NDIM)
      IP(N)=1
      DO 6 K=1,N
      IF(K.EQ.N) GO TO 5
      KP1=K+1
      M=K
      DO 1 I=KP1,N
      IF(ABS(A(I,K)).GT.ABS(A(M,K))) M=I
    1 CONTINUE
      IP(K)=M
      IF(M.NE.K) IP(N)=-IP(N)
      T=A(M,K)
      A(M,K)=A(K,K)
      A(K,K)=T
      IF(T.EQ.0.) GO TO 5
      DO 2 I=KP1,N
    2 A(I,K)=-A(I,K)/T
      DO 4 J=KP1,N
      T=A(M,J)
      A(M,J)=A(K,J)
      A(K,J)=T
      IF(T.EQ.0.) GO TO 4
      DO 3 I=KP1,N
    3 A(I,J)=A(I,J)+A(I,K)*T
    4 CONTINUE
    5 IF(A(K,K).EQ.0.) IP(N)=0
    6 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SOLVE(N,NDIM,A,B,IP)
      DIMENSION A(NDIM,NDIM),B(NDIM),IP(NDIM)
      IF(N.EQ.1) GO TO 9
      NM1=N-1
      DO 7 K=1,NM1
      KP1=K+1
      M=IP(K)
      T=B(M)
      B(M)=B(K)
      B(K)=T
      DO 7 I=KP1,N
    7 B(I)=B(I)+A(I,K)*T
      DO 8 KB=1,NM1
      KM1=N-KB
      K=KM1+1
      B(K)=B(K)/A(K,K)
      T=-B(K)
      DO 8 I=1,KM1
    8 B(I)=B(I) +A(I,K)*T
    9 B(1)=B(1)/A(1,1)
      RETURN
      END
```

```
#FILE (CHXL)TBL18 ON DTNSRDC
FILE  5(KIND=REMOTE,MAXRECSIZE=22)
FILE  6(KIND=PRINTER,MAXRECSIZE=132)
FILE  8(TITLE="BNSAL",KIND=DISK,FILETYPE=7)
FILE  9(KIND=REMOTE,MAXRECSIZE=22)
FILE  10(TITLE="TBLINP",KIND=DISK,FILETYPE=7)
FILE  11(TITLE="TBLSAVE",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
 $RESET FREE
      COMMON / /TITLE(12)
      COMMON /A/TTH(50),HB(50),THMB(50),BETB(50),DTA(50),
     1           RS(50),RTH(50),RD(50),CFF(50),XX(50),YY(50)
      COMMON /B2/ THNB(50),DTHS(50),DGS(50),DTNS(50),USR(50)
      COMMON /U/AN(10),ANP(10),ANPP(10),CN(10),CNP(10),GN(10),CNPP(10),
     1PAC
      COMMON /G/BN(25,10),WK(50),DK(50),TM(50),TMO
      COMMON / /TITL(3)
      COMMON /SAVE/GRU(50)
      DO 120 I=1,25
      DO 120 J=1,10
  120 BN(I,J)=0.0
      READ(8,1) (TITLE(J),J=1,12)
      WRITE(9,1) (TITLE(J),J=1,12)
      READ(8,405)N,KPK
      WRITE(9,405)N,KPK
      DO 13 J=1,N
      READ(8,3 ) (BN(I,J),I=1,KPK)
      WRITE(9,3) ( BN(I,J),I=1,KPK)
   13 CONTINUE
       WRITE(9,414)
       WRITE(9,415)
       WRITE(9,416)
       WRITE(9,417)
       WRITE(9,418)
       READ(5,412) NINPT
       GOTO (301,302,303) NINPT
  301 WRITE(9,419)
      STOP
  302 WRITE(9,400)
      READ(5,401) UO
      WRITE(9,401) UO
      WRITE(9,402)
      READ(5,401) BL
      WRITE(9,401) BL
      WRITE(9,409)
      READ(5,405) NSTAT
      WRITE(9,405) NSTAT
```

```
      WRITE(9,410)
      READ(5,3) S1,S2
      WRITE(9,3) S1,S2
      WRITE(9,404)
      READ(5,405) NTH
      WRITE(9,405) NTH
      WRITE(9,411)
      READ(5,401) (TTH(I),I=1,NTH)
      WRITE(9,401) (TTH(I),I=1,NTH)
      WRITE(9,403)
      DO 200 IN=1,NTH
      READ(5,406) HB(IN),THMB(IN),BETB(IN)
      WRITE(9,406) HB(IN),THMB(IN),BETB(IN)
  200 CONTINUE
      WRITE(9,407)
      READ(5,6) RE
      WRITE(9,6) RE
      WRITE(11,401) UO
      WRITE(11,401) BL
      WRITE(11,405) NSTAT
      WRITE(11,3) S1,S2
      WRITE(11,405) NTH
      WRITE(11,401) (TTH(I),I=1,NTH)
      DO 201 IN=1,NTH
      WRITE(11,406) HB(IN),THMB(IN),BETB(IN)
  201 CONTINUE
      WRITE(11,6) RE
      GO TO 304
  303 READ(10,401) UO
      READ(10,401) BL
      READ(10,405) NSTAT
      READ(10,3) S1,S2
      READ(10,405) NTH
      READ(10,401) (TTH(I),I=1,NTH)
      DO 202 IN=1,NTH
      READ(10,406) HB(IN),THMB(IN),BETB(IN)
  202 CONTINUE
      READ(10,6) RE
  304 WRITE(6,130)
      WRITE(6,110)
      WRITE(6,111)N,KPK,BL
      WRITE(6,113) NTH,NSTAT
      WRITE(6,12) S1,S2
      DO 121 J=1,N
  121 WRITE(6,112)(BN(I,J),I=1,KPK)
```

```
      DO 21 I=1,KPK
      DO 21 J=1,N
   21 BN(I,J)=BL*BN(I,J)
      DO 50 I=1,NTH
      DTA(I)=0.0
   50 CONTINUE
      DO 14 I=1,NTH
      G=1.535*((HB(I)-0.7)**(-2.715))+3.3
      E=-(G+HB(I))*(2./(HB(I)*(HB(I)+1.)*(HB(I)+2.)))
   14 THNB(I)=THMB(I)*E*BETB(I)
   20 WRITE(6,8)
      WRITE(6,1) (TITLE(J),J=1,12)
      RN=RE/2.
      N1=NSTAT+1
      DS=(S2-S1)/NSTAT
      ST=S1-DS
      CALL GLUG(N,KPK)
      DO 7 JJ=1,N1
      ST=ST+DS
      CALL UGH(ST,N,KPK,FAC,FACP,IENDS)
      WRITE(6,27) JJ,ST
      CALL AGH(FAC,FACP,N,NTH,JJ,DS,RN,ST)
    7 CONTINUE
  100 CONTINUE
      STOP
  500 WRITE(9,413)
      STOP
    1 FORMAT(12A6)
    2 FORMAT(2I10,2F10.5)
    3 FORMAT(7F10.5)
    5 FORMAT(2I10,2F10.5)
    6 FORMAT(F20.1)
    8 FORMAT(1H1)
    9 FORMAT(7F10.8)
   12 FORMAT(5H  S1=F10.5,4H S2=F10.5)
   27 FORMAT(//12H           S(I3,2H)=F8.4)
  110 FORMAT(/7H  INPUT/)
  111 FORMAT(4H  N=I3,5H KPK=I3,4H BL=F10.5)
  112 FORMAT(7(E12.5,5X))
  113 FORMAT(6H  NTH=I5,7H NSTAT=I5)
  130 FORMAT(1H1)
  400 FORMAT(20H U0=       1F10.5           )
  401 FORMAT(1F10.5)
  402 FORMAT(20H BL=       1F10.5           )
  403 FORMAT(38H   HB         THMB       BETB   3F10.5
```

```
404 FORMAT(15H NTH=     1I5      )
405 FORMAT(2I5)
406 FORMAT(3F10.5)
407 FORMAT(24H RE=           1F20.1        )
408 FORMAT(4H RE=)
409 FORMAT(24H NSTAT=          1I5           )
410 FORMAT(30H      S1        S2        7F10.5    )
411 FORMAT(25H TTH=         1F10.5               )
412 FORMAT(1I1)
413 FORMAT(25H END OF FILE REACHED         )
414 FORMAT(38H NINPT DENOTES THE TYPE OF INPUT              )
415 FORMAT(23H NINPT=1  CARD DATA      )
416 FORMAT(23H        2  REMOTE         )
417 FORMAT(23H        3  STORED DATA    )
418 FORMAT(//15H NINPT     1I1            )
419 FORMAT(43H PROVIDE READ STATEMENTS FOR CARDS AT 301
    END
```

```
      SUBROUTINE AGH(FAC,FACP,N,NTH,JJ,DS,RN,ST)
      COMMON /A/TTH(50),HB(50),THMB(50),BETB(50),DTA(50),
     1          RS(50),RTH(50),RD(50),CFF(50),XX(50),YY(50)
      COMMON /B2/ THNB(50),DTHS(50),DGS(50),DTNS(50),USR(50)
      COMMON /U/AN(10),ANP(10),ANPP(10),CN(10),CNP(10),GN(10),GN  10),
     1FAC
      COMMON /SAVE/GRU(50)
      DIMENSION PP(50),HH2(50),CKK1(50),CKK2(50)
      DIMENSION PPS(50),PHI(50),THH(50)
      DIMENSION RAD(50)
      DIMENSION CFL(50),CFWL(50),CFBL(50),CSL(50),PSIN( 5
      DO 1 I=1,NTH
      TH=TTH(I)
      X=0.0
      Y=0.0
      SUM1=0.0
      SUM2=0.0
      SUM3=0.0
      SUM4=0.0
      SUM5=0.0
      SUM6=0.0
      SUM7=0.0
      SUM8=0.0
      SUM9=0.0
      SUM10=0.0
      SUM11=0.0
      SUM12=0.0
      SUM13=0.0
      SUM14=0.0
      SUM15=0.0
      SUM17=0.0
      SUM18=0.0
      SUM19=0.0
      SUM20=0.0
      SUM21=0.0
      DO 23 J=1,N
      C1=2*J
      C4=3.-C1
      CS4=COS(C4*TH)
          SIN(C4*TH)
          =SUM1-C4*AN(J)*SS4
      SUM2=SUM2+C4*AN(J)*CS4
      SUM3=SUM3+ANP(J)*CS4
      SUM4=SUM4+ANP(J)*SS4
      SUM5=SUM5-(C4**2)*AN(J)*CS4
```

```
      SUM6=SUM6-(C4**2)*AN(J)*SS4
      SUM7=SUM7-C4*ANP(J)*SS4
      SUM8=SUM8+C4*ANP(J)*CS4
      SUM19=SUM19+ANPP(J)*CS4
      SUM20=SUM20+ANPP(J)*SS4
      X=X+AN(J)*CS4
 23   Y=Y+AN(J)*SS4
      DO 25 J=2,N
      C2=2*J-2
      SS2=SIN(C2*TH)
      CS2=COS(C2*TH)
      SUM9=SUM9+CN(J)*SS2
      SUM10=SUM10+CN(J)*CS2
      SUM11=SUM11+C2*CN(J)*CS2
      SUM12=SUM12-C2*CN(J)*SS2
      SUM13=SUM13-CNP(J)*CS2/C2
      SUM14=SUM14-CN(J)*CS2/C2
 52   SUM15=SUM15-CNPP(J)*CS2/C2
      SUM21=SUM21+GN(J)*SS2
      SUM17=SUM17+CNP(J)*SS2
 25   SUM18=SUM18+CNP(J)*CS2
      SUM10=CN(1)+SUM10
      SUM13 =SUM13+FAC+1.0
      SUM31=SUM13*SUM13
      SUM14=SUM14+FAC
      SUM18=SUM18+CNP(1)
      DE=SUM1**2+SUM2**2
131   DE2=DE**2
      HE=(SUM5*SUM2-SUM6*SUM1)**2
      U=(SUM9*SUM1+SUM10*SUM2)/DE
      V=(SUM9*SUM2-SUM10*SUM1)/DE
      W=SUM13-U*SUM3-V*SUM4-1.0
      DN=SUM1*SUM7+SUM2*SUM8
      US= (+SUM18*SUM2+SUM10*SUM8+SUM17*SUM1+SUM9*SUM7)/DE
      US=US-2.*DN*U/DE
      VS= (-SUM18*SUM1-SUM10*SUM7+SUM17*SUM2+SUM9*SUM8)/DE
      VS=VS-2.*DN*V/DE
      WS=FACP+SUM15-(US*SUM3+U*SUM19+VS*SUM4+V*SUM20)
      DTDS=2.0*SUM21/DE
      DM=SUM1*SUM5+SUM2*SUM6
      UTH= (SUM10*SUM6+SUM12*SUM2+SUM11*SUM1+SUM9*SUM5)/DE
      UTH=UTH-2.*DM*U/DE
      VTH= (-SUM12*SUM1-SUM10*SUM5+SUM11*SUM2+SUM9*SUM6)/DE
      VTH=VTH-2.*DM*V/DE
      WTH=SUM17-UTH*SUM3-U*SUM7-VTH*SUM4-V*SUM8
```

```
      G1=DE**(-0.5)
      HX=SUM3+DTDS*SUM1
      HY=SUM4+DTDS*SUM2
      PHI(I)=SUM14
      RAD(I)=(SUM5*SUM2**2-SUM6*SUM1*SUM2)**2+(SUM6*SUM1**2-SUM5
     1*SUM1*SUM2)**2
      RAD(I)=SQRT(RAD(I))
      RAD(I)=DE2/RAD(I)
      USS=U**2+V**2+2.0*W+1.0
      P=-USS+1.0
      PN=G1*(U*UTH+V*VTH+WTH)
      CK2=PN/USS
   53 PS=U*US+V*VS+WS+DTDS*PN/G1
      PPS(I)=PS/USS
  135 USS=SQRT(USS)
  200 THH(I)=TH
      F=-SUM9/SUM13
      FS=-SUM17/SUM13+SUM19*(FACP+SUM15)/SUM31
      FT=-SUM11/SUM13+SUM9*SUM17/SUM31
      GX=F*SUM3+SUM1
      GY=F*SUM4+SUM2
      GG=GX*GX+GY*GY+F*F
      HH=HX*HX+HY*HY+1.0
      HH=SQRT(HH)
      GXP=F*(SUM19*F+SUM3*FS+SUM7)+F*SUM7+SUM3*FT+SUM5
      GYP=F*(SUM20*F+SUM4*FS+SUM8)+F*SUM8+SUM4*FT+SUM6
      GZP=F*FS+FT
      RG=(GX*GXP+GY*GYP+F*GZP)/GG
       CK1=   (HX*(GXP-GX*RG)+HY*(GYP-GY*RG)+(GZP-F*RG))/(GG*HH)
      XX(I)=X
      YY(I)=Y
      PP(I)=P
      HH2(I)=DTDS
      USR(I)=USS
      CKK1(I)=CK1
      CKK2(I)=CK2
    1 CONTINUE
   14 WRITE(6,32)
      WRITE(6,31) (THH(I),XX(I), Y(I),PP(I),PPS(I),HH2(I),RAD(I),
     1CKK1(I),CKK2(I),PHI(I),I=1,NTH)
      IF(JJ.EQ.1) GO TO 3
      DO 4 I=1,NTH
      GRU(I)=(PHI(I)-GRU(I))/USR(I)
      THMB(I)=THMB(I)+1.*GRU(I)*DTHS(I)
      G=1.535*((HB(I)-0.7)**(-2.715))+3.3
```

```
      G=G+GRU(I)*DGS(I)
      THNB(I)=THNB(I)+GRU(I)*DTNS(I)
      IF(G.LE.3.3) GO TO 501
      HB(I)=(1.535/(G-3.3))**0.3646+0.7
      H=HB(I)
      E=-(G+H)*(2./(H*(H+1.)*(H+2.)))
      BETB(I)=1.*THNB(I)/(THMB(I)*E)
    4 CONTINUE
    3 DO 2 I=1,NTH
      H=HB(I)
      THM=THMB(I)
      BET=BETB(I)
      USO=USR(I)
      PSO=PPS(I)
      CK1=CKK1(I)
      CK2=CKK2(I)
      CALL BLAY(RN,USO,PSO,CK1,CK2,H.THM,BET,CF,DTH,DG,DTN,THN)
      RTH(I)=RN*USO*THM
      RD(I)=RTH(I)*H
      DTA(I)=H*THMB(I)
      CFL(I)=2000.*CF
      CF=CFL(I)
      CALL FCF(SUM1,SUM2,SUM3,SUM4,DTDS,BET,CF.CFWW,CFBB,CS3)
      CFWL(I)=CFWW
      CFBL(I)=CFBB
      CSL(I)=CS3
      GRU(I)=FHI(I)
      TTH(I)=TTH(I)+DS*HH2(I)
      DTHS(I)=DTH
      DGS(I)=DG
      DTNS(I)=DTN
    2 CONTINUE
      WRITE(6,40) JJ,ST
      WRITE(6,34)
      WRITE(6,35) (HB(I),THMB(I),DTA(I),BETB(I),CFL(I),RS(I),RTH(I),
     1CFWL(I),CFBL(I),CSL(I),I=1,NTH)
      RETURN
  501 WRITE(9,406)
      WRITE(9,407) G
      WRITE(9,405) JJ,ST
      STOP
   31 FORMAT(6F12.5,E12.5,3F12.5)
   32 FORMAT(//120H        TH          X          Y          CP
     1  PSI          DTDS          R          K1          K2          PHI
     1 //)
```

```
   34 FORMAT(/119H       H             TH             DE             BETA
     1 CF            RS            RTH           CFW           CFB           [illegible]
     2/)
   35 FORMAT(5F12.5,1H ,E11.4,1H ,E11.4,1H ,3F12.5)
   40 FORMAT(//25H     BOUNDARY LAYER AT S(I3,2H)=F8.4)
  405 FORMAT(19H FAILED AT STA NO =,1I4,3H S=,1F8.5)
  406 FORMAT(46H G IS LESS THAN 3.3 CANNOT CONTINUE IN AGH
  407 FORMAT(3H G=,1E15.5)
      END
#
```

```
      SUBROUTINE BLAY(RN,USO,PSO,CKO1,CKO2,H,THM,BET,CF,DTH,DG,DTN,THN)
      G=1.535*((H-0.7)**(-2.715))+3.3
      F=0.0306*((G-3.)**(-0.653))
      THN2=THN**2
      RT=RN*USO*THM
      C=ALOG(RT)
      C2=C*C
      C3=C2*C
      A=0.019521-0.386768*C+0.028345*C2-0.000701*C3
      B=0.191511-0.834891*C+0.062588*C2-0.001953*C3
      CF=0.5*EXP(A*H+B)
   20 DTH=CF+THM*(CKO1-(2.+H)*PSO)
      DG=F/THM+G*(CKO1-PSO-DTH/THM)
      DTN=BET*CF+2.*THN*(CKO1-PSO)-THM*(1.+H)*CKO2
   10 RETURN
      END
```

```
      SUBROUTINE FCF(SUM1,SUM2,SUM3,SUM4,DTDS,BET,CF,CFWW,CFBB,CS3)
      X2=SUM1*SUM1+SUM2*SUM2
      Y=SUM1*SUM4-SUM2*SUM3
      Y2=Y*Y
      D=SQRT(X2+Y2)
      XN1=SUM2/D
      XN2=-SUM1/D
      XN3=Y/D
      S1=SUM3+SUM1*DTDS
      S2=SUM4+SUM2*DTDS
      SD=SQRT(S1*S1+S2*S2+1.)
      S1=S1/SD
      S2=S2/SD
      S3=1./SD
      B1=S2*XN3-S3*XN2
      B2=S3*XN1-S1*XN3
      B3=S1*XN2-S2*XN1
      CS1=CF*(S1+BET*B1)
      CS2=CF*(S2+BET*B2)
      CS3=CF*(S3+BET*B3)
      IF(SUM2) 1,2,1
    2 W1=0.
      W3=1.0
      GO TO 3
    1 W1=SUM3- SUM1*SUM4/SUM2
      WD=SQRT(W1*W1+1.)
      W1=W1/WD
      W3=1./WD
    3 IF(SUM1) 4,5,4
    5 D2=0.
      D3=1.0
      GO TO 6
    4 D2=SUM4-SUM2*SUM3/SUM1
      WD=SQRT(D2*D2+1.)
      D2=D2/WD
      D3=1./WD
    6 CFWW=W1*CS1+W3*CS3
      CFBB=D2*CS2+D3*CS3
      RETURN
      END
```

```
      SUBROUTINE GLUG(N,M)
      COMMON /G/BN(25,10),WK(50),DK(50),TM(50),TMO
      DIMENSION AMK(10,50)
      M2=2*M-1
      M3=M2-3
      DO 1 I=1,N
      DO 2 J=2,M2
      SUM=0.0
      IF(J-M) 3,3,4
    3 DO 5 L=1,J
      C=L-1
      L1=J+1-L
    5 SUM=SUM+C*(BN(L1,I)*BN(L,I))
      GO TO 7
    4 J1=J+1-M
      DO 6 L=J1,M
      C=L-1
      L1=J+1-L
    6 SUM=SUM+C*(BN(L1,I)*BN(L,I))
    7 AMK(I,J)=SUM
      WK(J)=0.0
    2 CONTINUE
    1 CONTINUE
      DO 29 J=3,M2
      J2=J-2
      SUM=0.0
      DO 30 L=1,J2
      C=L
      C=1./C
   30 SUM=SUM+C
   29 WK(J)=SUM
      DO 8 J=2,M2
      SUM=0.0
      DO 9 I=1,N
      C=3-2*I
    9 SUM=SUM+C*AMK(I,J)
    8 DK(J)=SUM
      DO 27 LM=1,M3
      SUM=0.0
      LMM=LM+3
      UN=1.
      DO 28 LK=LMM,M2
      KM=LK-LM-2
      XMK=KM
      UN=-1.*UN
```

```
28 SUM=SUM+DK(LK)*(1.+UN)/XMK
27 TM(LM)=SUM
   TMO=0.0
   DO 10 J=3,M2
   AJ=J-2
   U=(-1.)**(J-2)
10 TMO=TMO+DK(J)*(1.+U)/AJ
   RETURN
   END
#
```

```
      SUBROUTINE UGH(ST,N,M,FAC,FACP,IENDS)
      COMMON /G/BN(25,10),WK(50),DK(50),IM(50),IMO
      COMMON /U/AN(10),ANP(10),ANPP(10),CN(10),CNP(10),GN(10),CNPP(10),
     1FAC
      DIMENSION ANPPP(10),SP(50)
      M2=2*M-1
      M3=M2-3
      SP(1)=1.0
      DO 11 L=2,M2
   11 SP(L)=SP(L-1)*ST
      DO 12 J=1,N
      AN(J)=0.0
      ANP(J)=0.0
      ANPP(J)=0.0
      ANPPP(J)=0.0
      DO 13 L=1,M
   13 AN(J)=AN(J)+BN(L,J)*SP(L)
      DO 14 L=2,M
      C=L-1
   14 ANP(J)=ANP(J)+BN(L,J)*SP(L-1)*C
      DO 15 L=3,M
      C=(L-2)*(L-1)
   15 ANPP(J)=ANPP(J)+BN(L,J)*SP(L-2)*C
      DO 22 L=4, M
      C=(L-3)*(L-2)*(L-1)
   22  ANPPP(J)=ANPPP(J)+C*BN(L,J)*SP(L-3)
   12 CONTINUE
      GN(1)=0.0
      CN(1)=0.0
      CNP(1)=0.0
      CNPP(1)=0.0
      DO 16 L=1,N
      C=3-2*L
      CN(1)=CN(1)+ANP(L)*AN(L)*C
      CNPP(1)=CNPP(1)+3.0*C*ANP(L)*ANPP(L)+C*AN(L)*ANPPP(L)
   16 CNP(1)=CNP(1)+(ANPP(L)*AN(L)+ANP(L)**2)*C
   27 DO 17 J=2,N
      CN(J)=0.0
      CNP(J)=0.0
      CNPP(J)=0.0
      LU=N-J+1
      DO 18 L=1,LU
      C=3-2*L
      LJ=J+L-1
      CN(J)=CN(J)+ANP(LJ)*AN(L)*C
```

```
      CNPP(J)=CNPP(J)+2.0*C*ANP(L)*ANPP(LJ)+C*ANPP(L)*ANP(LJ)
     1+C*AN(L)*ANPPP(LJ)
   18 CNP(J)=CNP(J)+(ANPP(LJ)*AN(L)+ANP(LJ)*ANP(L))*C
      GN(J)=CN(J)
      DO 19 L=J,N
      C=3-2*L
      LJ=L+1-J
      CN(J)=CN(J)+ANP(LJ)*AN(L)*C
      CNPP(J)=CNPP(J)+2.0*C*ANP(L)*ANPP(LJ)
     1+C*ANPP(L)*ANP(LJ)+C*AN(L)*ANPPP(LJ)
   19 CNP(J)=CNP(J)+(ANPP(LJ)*AN(L)+ANP(LJ)*ANP(L))*C
   17 CONTINUE
      SUM1=0.0
      SUM2=0.0
      DO 20 L=3,M2
      CL=L-2
   20 SUM1=SUM1+CL*DK(L)*WK(L)*SP(L-2)
      DO 21 L=1,M3
      CL=L
   21 SUM2=SUM2+CL*TM(L)*SP(L)
      SUM3=0.0
      DO 23 L=4,M2
      CL=(L-2)*(L-3)
   23 SUM3=SUM3+CL*DK(L)*WK(L)*SP(L-3)
      SUM4=0.0
      DO 24 L=2,M3
      CL=L*(L-1)
   24 SUM4=SUM4+CL*TM(L)*SP(L-1)
      PS=4.-4.*SP(3)
      BOP=-0.5*CNP(1)*ALOG(PS)+4.*SP(2)*CN(1)/PS+SUM1-0.5*SUM2
      FAC=BOP+CNP(1)*ALOG(AN(1))+CN(1)*ANP(1)/AN(1)
      BOPP=-0.5*CNPP(1)*ALOG(PS)+4.*(CN(1)+2.*SP(2)*CNP(1))/PS
     1+32.*SP(3)*CN(1)/(PS*PS)+SUM3-0.5*SUM4
   41 FACP=BOPP+CNPP(1)*ALOG(AN(1))+(2.0*CNP(1)*ANP(1)+CN(1)*ANPP(1))
     1/AN(1)-CN(1)*(ANP(1)/AN(1))**2
      BO=0.0
      DO 25 J=3,M2
   25 BO=BO+WK(J)*DK(J)*SP(J-1)
      DO 26 J=1,M3
   26 BO=BO-0.5*TM(J)*SP(J+1)
      BO=BO-0.5*TMO
      FAC=CN(1)*ALOG(AN(1))+BO+ST-0.5*CN(1)*ALOG(PS)
      RETURN
      END
```

```
FILE  5(KIND=REMOTE,MAXRECSIZE=22)
FILE  6(KIND=PRINTER,MAXRECSIZE=132)
FILE  7(TITLE="TBL3DOPT",KIND=DISK,PROTECTIO=SAVE,MAXRECSIZE=22)
FILE  8(TITLE="BNSAL",KIND=DISK,FILETYPE=7)
FILE  9(KIND=REMOTE,MAXRECSIZE=22)
FILE  10(TITLE="TBL3DINP",KIND=DISK,FILETYPE=7)
FILE  11(TITLE="TBLSAVE",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
 $RESET FREE
C  PROGRAM TURBULENT BOUNDARY LAYER COORDINATE SYSTEM, BODY
C  GEOMETRY, AND POTENTIAL
      COMMON /A2/ BN(25,10)
      COMMON /A3/ WK(50),DK(50),TM(50),CN(10),CNP(10),GN(10)
      COMMON /A4/ AN(10),ANP(10),ANPP(10)
      COMMON /A6/ TTH(50)
      COMMON /B5/ XA(100,50),YA(100,50),SV(100,50),CF(100,50)
      COMMON / / TITLE(12),TITL(3)
      DO 21 I=1,25
      DO 21 J=1,10
   21 BN(I,J)=0.0
      DO 22 I=1,50
      WK(I)=0.0
      DK(I)=0.0
   22 TM(I)=0.0
      READ(8,1) (TITLE(J),J=1,12)
      WRITE(9,1) (TITLE(J),J=1,12)
      WRITE(7,1) (TITLE(J),J=1,12)
      READ(8,405)N,KPK,BL
      WRITE(9,405)N,KPK,BL
      DO 13 J=1,N
      READ(8,3 ) (BN(I,J),I=1,KPK)
      WRITE(9,3) ( BN(I,J),I=1,KPK)
   13 CONTINUE
       WRITE(9,414)
       WRITE(9,415)
       WRITE(9,416)
       WRITE(9,417)
       WRITE(9,418)
       READ(5,412) NINPT
       GOTO (301,302,303) NINPT
  301 WRITE(9,419)
      STOP
  302 WRITE(9,400)
      WRITE(9,420)
      READ (5,412) LBL
      IF(LBL.NE.1) GOTO 210
```

```
      WRITE(9,421)
      READ(5,401)BL
      WRITE(9,401)BL
  210 WRITE(9,422)
      READ (5,401)START
      WRITE(9,401)START
      WRITE(9,423)
      READ (5,401)SFINIS
      WRITE(9,401)SFINIS
      WRITE(9,424)
      READ (5,405)NSTAT
      WRITE(9,405)NSTAT
      WRITE(9,425)
      READ (5,405)NTH
      WRITE(9,405)NTH
      WRITE(11,401) BL
      WRITE(11,401) START
      WRITE(11,401) SFINIS
      WRITE(11,405) NSTAT
      WRITE(11,405) NTH
      GO TO 304
  303 READ (10,401)BL
      READ (10,401) START
      READ (10,401) SFINIS
      READ (10,405) NSTAT
      READ (10,405) NTH
  304 CONTINUE
      WRITE(7,2) N,KPK,BL
      WRITE(7,5) START,SFINIS,NSTAT,NTH
      TTH(1)=0.0
      PI02= -1.57079632679
      EN=NTH-1
      DT=PI02/EN
      DO 200 I=2,NTH
  200 TTH(I)=TTH(I-1)+DT
      WRITE(9,3)(TTH(I),I=1,NTH)
   20 WRITE(6,8)
      WRITE(6,1) (TITLE(J),J=1,12)
      DO 4 I=1,KPK
      DO 4 J=1,N
    4 BN(I,J)=BL+BN(I,J)
      EN=NSTAT-1
      DS=(SFINIS-START)/EN
      DSO=DS
      ID=-START/DS+1.
```

```
      DO 23 I=1,10
      AN(I)=0.0
      ANP(I)=0.0
      ANPP(I)=0.0
      CN(I)=0.0
      CNP(I)=0.0
   23 GN(I)=0.0
      CALL GLUG(N,KPK)
      ST=START-DS
      DO 7 JJ=1,NSTAT
      ID=0
      INDEX=0
    6 ST=ST+DS
      CALL AGH(FAC,N,NTH,JJ,DS,ST,ID,INDEX,KPK)
      DS=DSO
  500 CONTINUE
    7 CONTINUE
    1 FORMAT(12A6)
    2 FORMAT(2I10,F10.5)
    3 FORMAT(7F10.5)
    5 FORMAT(2F10.5,2I10)
    8 FORMAT(1H1)
   70 FORMAT(3A10)
  400 FORMAT(30H DO YOU WISH TO ENTER NEW BL?    )
  401 FORMAT(1F10.5)
  402 FORMAT(20H BL=       1F10.5           )
  403 FORMAT(38H   HB          THMB       BETB  3F10.5      )
  404 FORMAT(15H NTH=     1I5      )
  405 FORMAT(2I5,1F10.5)
  406 FORMAT(3F10.5)
  407 FORMAT(24H RE=           1F20.1        )
  408 FORMAT(4H RE=)
  409 FORMAT(24H NSTAT=          1I5          )
  410 FORMAT(30H     S1        S2        7F10.5   )
  411 FORMAT(25H TTH=         1F10.5              )
  412 FORMAT(1I1)
  413 FORMAT(25H END OF FILE REACHED        )
  414 FORMAT(38H NINPT DENOTES THE TYPE OF INPUT            )
  415 FORMAT(23H NINPT=1  CARD DATA      )
  416 FORMAT(23H        2  REMOTE        )
  417 FORMAT(23H        3  STORED DATA   )
  418 FORMAT(15H NINPT     1I1            )
  419 FORMAT(43H PROVIDE READ STATEMENTS FOR CARDS AT 301        )
  420 FORMAT(27H IF YES TYPE 1 NO TYPE 0     )
  421 FORMAT(15H BL=     1F10.5   )
```

AD-A099 108 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 20/4
A USER'S MANUAL FOR THE DTNSRDC MOMENTUM INTEGRAL BOUNDARY-LAYE--ETC(U)
DEC 80 T J LANGAN, C VON KERCZEK
UNCLASSIFIED DTNSRDC/SPD-0972-01 NL

2 of 2
AD
A099 108

END
DATE
FILMED
8-81
DTIC

```
 422 FORMAT(20H START=      1F10.5    )
 423 FORMAT(20H SFINIS=      1F10.5   )
 424 FORMAT(20H NSTAT=       1I3      )
 425 FORMAT(20H NTH=         1I3       )
     END
#
```

```
      SUBROUTINE GLUG(N,M)
      COMMON /A2/ BN(25,10)
      COMMON /A3/ WK(50),DK(50),TN(50),CN(10),CNP(10),GN(10)
      DIMENSION AMK(10,50)
      M2=2*M-1
      M3=2*M-4
      DO 31 I=1,10
      DO 31 J=1,50
   31 AMK(I,J)=0.0
      DO 1 I=1,N
      DO 2 J=2,M2
      SUM=0.0
      IF(J-M) 3,3,4
    3 DO 5 L=1,J
      C=L-1
      L1=J+1-L
    5 SUM=SUM+C*(BN(L1,I)*BN(L,I))
      GO TO 7
    4 J1=J+1-M
      DO 6 L=J1,M
      C=L-1
      L1=J+1-L
    6 SUM=SUM+C*(BN(L1,I)*BN(L,I))
    7 AMK(I,J)=SUM
      WK(J)=0.0
    2 CONTINUE
    1 CONTINUE
      DO 29 J=3,M2
      J2=J-2
      SUM=0.0
      DO 30 L=1,J2
      C=L
      C=1./C
   30 SUM=SUM+C
   29 WK(J)=SUM
      DO 8 J=2,M2
      SUM=0.0
      DO 9 I=1,N
      C=3-2*I
    9 SUM=SUM+C*AMK(I,J)
    8 DK(J)=SUM
      DO 27 LM=1,M3
      SUM=0.0
      LMM=LM+3
      UN=1.
```

```
      DO 28 LK=LMM,M2
      KM=LK-LM-2
      XMK=KM
      UN=-1.*UN
   28 SUM=SUM+DK(LK)*(1.+UN)/XMK
   27 TM(LM)=SUM
      RETURN
      END
#
```

```
      SUBROUTINE POT(ST,N,M,FAC)
      COMMON /A4/ AN(10),ANP(10),ANPP(10)
      COMMON /A3/ WK(50),DK(50),TM(50),CN(10),CNP(10),GN(10)
      COMMON /C1/ SP(50)
      M2=2*M-1
      M3=M2-3
      GN(1)=0.0
      CN(1)=0.0
      CNP(1)=0.0
      DO 16 L=1,N
      C=3-2*L
      CN(1)=CN(1)+ANP(L)*AN(L)*C
   16 CNP(1)=CNP(1)+(ANPP(L)*AN(L)+ANP(L)**2)*C
      DO 17 J=2,N
      CN(J)=0.0
      CNP(J)=0.0
      LU=N-J+1
      DO 18 L=1,LU
      C=3-2*L
      LJ=J+L-1
      CN(J)=CN(J)+ANP(LJ)*AN(L)*C
   18 CNP(J)=CNP(J)+(ANPP(LJ)*AN(L)+ANP(LJ)*ANP(L))*C
      GN(J)=CN(J)
      DO 19 L=J,N
      C=3-2*L
      LJ=L+1-J
      CN(J)=CN(J)+ANP(LJ)*AN(L)*C
   19 CNP(J)=CNP(J)+(ANPP(LJ)*AN(L)+ANP(LJ)*ANP(L))*C
   17 CONTINUE
      SUM1=0.0
      SUM2=0.0
      DO 20 L=3,M2
      CL=L-2
   20 SUM1=SUM1+CL*DK(L)*WK(L)*SP(L-2)
      DO 21 L=1,M3
      CL=L
   21 SUM2=SUM2+CL*TM(L)*SP(L)
      PS=4.-4.*SP(3)
      BOP=4.*SP(2)*CN(1)/PS
      BOP=BOP-0.5*CNP(1)*ALOG(PS)+SUM1-0.5*SUM2
      FAC=BOP+CNP(1)*ALOG(AN(1))+CN(1)*ANP(1)/AN(1)
      RETURN
      END
#
```

```
      SUBROUTINE BODY(ST,N,M)
      COMMON /A2/ BN(25,10)
      COMMON /A4/ AN(10),ANP(10),ANPP(10)
      COMMON /C1/ SP(50)
      M2=2*M-1
      M3=2*M-4
      M4=M2+1
      SP(1)=1.0
      DO 11 L=2,M2
   11 SP(L)=SP(L-1)*ST
      DO 22 L=M4,50
   22 SP(L)=0.0
      DO 12 J=1,N
      AN(J)=0.0
      ANP(J)=0.0
      ANPP(J)=0.0
      DO 13 L=1,M
   13 AN(J)=AN(J)+BN(L,J)*SP(L)
      DO 14 L=2,M
      C=L-1
   14 ANP(J)=ANP(J)+BN(L,J)*SP(L-1)*C
      DO 15 L=3,M
      C=(L-2)*(L-1)
   15 ANPP(J)=ANPP(J)+BN(L,J)*SP(L-2)*C
   12 CONTINUE
      RETURN
       END
#
```

```
      SUBROUTINE BDER(DSDT,S,N,KPK,TH)
      COMMON /A4/ AN(10),ANP(10),ANPP(10)
      CALL BODY(S,N,KPK)
      XS=0.0
      YS=0.0
      XT=0.0
      YT=0.0
      DO 1 J=1,N
      C=3-2*J
      CC=COS(C*TH)
      SS=SIN(C*TH)
      XS=XS+CC*ANP(J)
      YS=YS+SS*ANP(J)
      XT=XT-SS*C*AN(J)
    1 YT=YT+CC*C*AN(J)
      DSDT=-(XS*XT+YS*YT)/(XS*XS+YS*YS+1.)
      RETURN
      END
```

```
      SUBROUTINE AGH(FAC,N,NTH,JJ,DS,ST,ID,INDEX,KPK)
      COMMON /A3/ WK(50),DK(50),TM(50),CN(10),CNP(10),GN(10)
      COMMON /A4/ AN(10),ANP(10),ANPP(10)
      COMMON /A6/ TTH(50)
      COMMON /B5/ XA(100,50),YA(100,50),SV(100,50),CP(100,50)
      DIMENSION SS(50)
      WRITE(6,27) JJ,ST
      WRITE(7,427) JJ,ST
      SS(1)=ST
      DO 1 I=1,NTH
      IF(I.GT.1) GO TO 2
      TH=TTH(1)
      S=SS(1)
      CALL BDER(DSDT,S,N,KPK,TH)
      GO TO 3
    2 TH=TTH(I-1)
      S=SS(I-1)
      CALL BDER(DSDT,S,N,KPK,TH)
      S1=SS(I-1)+(TTH(I)-TTH(I-1))*DSDT
      TH=TTH(I)
      CALL BDER(DSDTC,S1,N,KPK,TH)
      SS(I)=SS(I-1)+0.5*(TTH(I)-TTH(I-1))*(DSDTC+DSDT)
      S=SS(I)
    3 CALL POT(S,N,KPK,FAC)
      X=0.0
      Y=0.0
      SUM1=0.0
      SUM2=0.0
      SUM3=0.0
      SUM4=0.0
      SUM5=0.0
      SUM6=0.0
      SUM7=0.0
      SUM8=0.0
      SUM9=0.0
      SUM10=0.0
      SUM11=0.0
      SUM12=0.0
      SUM13=0.0
      SUM14=0.0
      SUM15=0.0
      SUM16=0.0
      SUM17=0.0
      SUM18=0.0
      SUM19=0.0
```

```
      SUM20=0.0
      SUM21=0.0
      DO 23 J=1,N
      C1=2*J
      C2=1.-C1
      C3=2.-C1
      C4=3.-C1
      CS1=COS(C1*TH)
      CS2=COS(C2*TH)
      CS3=COS(C3*TH)
      CS4=COS(C4*TH)
      SS2=SIN(C2*TH)
      SS3=SIN(C3*TH)
      SS4=SIN(C4*TH)
      SUM1=SUM1+CNP(J+1)*CS1/C1
      SUM2=SUM2+CN(J)*CS2
      SUM3=SUM3+CN(J)*SS2
      SUM4=SUM4+ANP(J)*CS4
      SUM5=SUM5+ANP(J)*SS4
      SUM6=SUM6+AN(J)*CS3*C4
      SUM7=SUM7+AN(J)*SS3*C4
      SUM8=SUM8+ANPP(J)*CS4
      SUM9=SUM9+ANPP(J)*SS4
      SUM10=SUM10-AN(J)*C4*SS4
      SUM11=SUM11+AN(J)*C4*CS4
      SUM12=SUM12-ANP(J)*C4*SS4
      SUM13=SUM13+ANP(J)*C4*CS4
      SUM14=SUM14+CNP(J)*CS2
      SUM15=SUM14+CNP(J)*SS2
      SUM16=SUM16+C2*CN(J)*CS2
      SUM17=SUM17-C2*CN(J)*SS2
      SUM18=SUM18+C4*ANP(J)*CS3
      SUM19=SUM19+C4*ANP(J)*SS3
      SUM20=SUM20-C4*C3*AN(J)*SS3
      SUM21=SUM21+C4*C3*AN(J)*CS3
      X=X+AN(J)*CS4
   23 Y=Y+AN(J)*SS4
      DE=SUM6**2+SUM7**2
      DEI=1.0/DE
      U=(SUM2*SUM6+SUM3*SUM7)/DE
      V=(SUM2*SUM7-SUM3*SUM6)/DE
      W=-(U*SUM4+V*SUM5)-SUM1+FAC
      P=-2.*W-U**2-V**2
      XYSS=SUM6*SUM18+SUM7*SUM19
      XYST=SUM6*SUM20+SUM7*SUM21
```

```
      US=(SUM14*SUM6+SUM2*SUM18+SUM15*SUM7+SUM3*SUM19)
      US=(US-2.0*U*XYSS)*DEI
      UT=(SUM17*SUM6+SUM2*SUM20+SUM16*SUM7+SUM3*SUM21)
      UT=(UT-2.0*U*XYST)*DEI
      VS=SUM14*SUM7+SUM2*SUM19-SUM15*SUM6-SUM3*SUM18
      VS=(VS-2.0*V*XYSS)*DEI
      VT=SUM17*SUM7+SUM2*SUM21-SUM16*SUM6-SUM3*SUM20
      VT=(VT-2.0*V*XYST)*DEI
      H=SUM4*SUM4+SUM5*SUM5+1.0
      H2=SQRT(H)
      H2I=1.0/H2
      H3=H*H2
      F=-(SUM10*SUM4+SUM11*SUM5)/H
      G=(SUM10+SUM4*F)**2+(SUM11+SUM5*F)**2+F*F
      G=SQRT(G)
      GI=1.0/G
      DUDLP=H2I*US
      DVDLP=H2I*VS
      DUDLP=GI*(F*US+UT)
      DVBLT=GI*(F*VS+VT)
      D=SUM10*SUM10+SUM11*SUM11+(SUM10*SUM5-SUM11*SUM4)**2
      D=SQRT(D)
      EN1=SUM11/D
      EN2=-SUM10/D
      EN3=(SUM10*SUM5-SUM11*SUM4)/D
      ET1=(SUM5*EN3-EN2)/H2
      ET2=(EN1-SUM4*EN3)/H2
      ET3=(SUM4*EN2-SUM5*EN1)/H2
      EP1=SUM4*H2I
      EP2=SUM5*H2I
      EP3=H2I
C     COMPUTATION OF XKT THE GEODESIC CURVATURE OF THE PHI COORDINATE
      HD=(SUM4*SUM8+SUM5*SUM9)/H3
      EP1S= SUM8*H2I-SUM4*HD
      EP2S=SUM9*H2I-SUM5*HD
      EP3S = -HD
      XKT = (ET1*EP1S+ET2*EP2S+ET3*EP3S)*H2I
C     COMPUTATION OF XKP GEODESIC CURVATURE OF THATA COORDINATE LINES
      HD=(SUM4*SUM12+SUM5*SUM13)/H3
      EP1T = SUM12*H2I-SUM4*HD
      EP2T =SUM13*H2I-SUM5*HD
      EP3T = -HD
      GI=1.0/G
      P1LT= GI*(F*EP1S+EP1T)
      P2LT= GI*(F*EP2S+EP2T     )
```

```
      P3LT =GI*(F*EP3S+EP3T)
      E1=P2LT*EN3-P3LT*EN2
      E2=P3LT*EN1-P1LT*EN3
      E3=P1LT*EN2-P2LT*EN1
      XKP=(SUM4*E1+SUM5*E2+E3)*H2I
C     COMPUTATION OF UP,UT, AND UN VELOCITIES IN ORTHOGONAL COORDINATES.
      W=1.0    +W
      UN=EN1*U+EN2*V+EN3*W
      UT=ET1*U+ET2*V+ET3*W
      UP=EP1*U+EP2*V+EP3*W
       H=H2
      WRITE(6,408) 1,X,Y,TH,U,V
      WRITE(7,408) 1,X,Y,TH,U,V
      WRITE(6,408) 2,W,UP,UT,UN,SS(I)
      WRITE(7,408) 2,W,UP,UT,UN,SS(I)
      WRITE(6,408)3,XKP,XKT,H,F,G
      WRITE(7,408)3,XKP,XKT,H,F,G
    1 CONTINUE
  500 RETURN
   27 FORMAT(//7H     S(I2,2H)=F9.6)
  400 FORMAT(8E15.7)
  408 FORMAT(1I3,5E15.7)
  427 FORMAT(1I3,1F9.6)
      END
```

```
#FILE (CHXL)CBLGEO ON DTNSRDC
FILE  5(KIND=REMOTE,MAXRECSIZE=22)
FILE  7(TITLE="CBL3DOPT",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
FILE  8(TITLE="BNSAL",KIND=DISK,FILETYPE=7)
FILE  9(KIND=REMOTE,MAXRECSIZE=22)
FILE  10(TITLE="CBL3DINP",KIND=DISK,FILETYPE=7)
FILE  11(TITLE="CBLTAPE",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
 $RESET FREE
C    PROGRAM CBL
      COMMON /A2/ BN(25,10)
      COMMON /A3/ WK(50),DK(50),TM(50),CN(10),CNP(10),GN(10)
      COMMON /A4/ AN(10),ANP(10),ANPP(10)
      COMMON /A6/ TTH(50)
      COMMON /B5/ XA(100,50),YA(100,50),SV(100,50),CP(100,50)
      COMMON / / TITLE(12),TITL(3)
      DIMENSION S(50),X(50,50),Y(50,50)
      DIMENSION SE(500),XE(500),YE(500),THE(500)
      DO 21 I=1,25
      DO 21 J=1,10
   21 BN(I,J)=0.0
      DO 22 I=1,50
      WK(I)=0.0
      DK(I)=0.0
   22 TM(I)=0.0
      READ(8,1) (TITLE(J),J=1,12)
      WRITE(9,1) (TITLE(J),J=1,12)
      WRITE(7,1) (TITLE(J),J=1,12)
      READ(8,405)N,KPK,BL
      WRITE(9,405)N,KPK,BL
      DO 13 J=1,N
      READ(8,3 ) (BN(I,J),I=1,KPK)
      WRITE(9,3) ( BN(I,J),I=1,KPK)
   13 CONTINUE
       WRITE(9,414)
       WRITE(9,415)
       WRITE(9,416)
       WRITE(9,417)
       WRITE(9,418)
       READ(5,412) NINPT
       GOTO (301,302,303) NINPT
  301 WRITE(9,419)
      STOP
  302 WRITE(9,400)
      WRITE(9,420)
      READ (5,412) LBL
      IF(LBL.NE.1) GOTO 210
      WRITE(9,421)
      READ(5,401)BL
```

```
      WRITE(9,401)BL
  210 START=-1.000
      SFINIS=1.000
      WRITE(9,424)
      READ (5,429)NSTAT
      WRITE(9,405)NSTAT
      WRITE(9,425)
      READ (5,429)NTH
      WRITE(9,405)NTH
      WRITE(11,401) BL
      WRITE(11,401) START
      WRITE(11,401) SFINIS
      WRITE(11,405) NSTAT
      WRITE(11,405) NTH
      GO TO 304
  303 READ (10,401)BL
      READ (10,401) START
      READ (10,401) SFINIS
      READ (10,405) NSTAT
      READ (10,405) NTH
  304 CONTINUE
      WRITE(7,2) N,KPK,BL
      WRITE(7,5) START,SFINIS,NSTAT,NTH
      TTH(1)=0.0
      PIO2= -1.57079632679
      EN=NTH-1
      DT=PIO2/EN
      DO 200 I=2,NTH
  200 TTH(I)=TTH(I-1)+DT
      WRITE(9,3)(TTH(I),I=1,NTH)
      DO 4 I=1,KPK
      DO 4 J=1,N
    4 BN(I,J)=BL*BN(I,J)
      EN=NSTAT-1
      DS=(SFINIS-START)/EN
      DSO=DS
      KD=-START/DS+1.
      DO 23 I=1,10
      AN(I)=0.0
      ANP(I)=0.0
      ANPP(I)=0.0
      CN(I)=0.0
      CNP(I)=0.0
   23 GN(I)=0.0
      ST=START-DS
      NTHMO=NTH-1
      S(1)=ST
      DO 201 I=1,NSTAT
      S(I)=S(I)+DS
      DO 211 J=1,NTH
      CALL CORD(N,S(I),TTH(J),X(I,J),Y(I,J),KPK)
```

```
  211 CONTINUE
      S(I+1)=S(I)
  201 CONTINUE
      DO 204 J=1,NTH
      X(1,J)=0.0
  204 X(NSTAT,J)=0.0
      NSMO=NSTAT-1
      NTHMO=NTH-1
      DO 202 I=1,NSMO
      SSE=S(I)+0.5*DS
      DO 202 J=1,NTHMO
      MP=J+NTHMO*(I-1)
      SE(MP)=SSE
      THE(MP)=TTH(J)+0.5*DT
      CALL CORD(N,SE(MP),THE(MP),XE(MP),YE(MP),KPK)
  202 CONTINUE
      DO 203 I=1,NSMO
      DO 203 J=1,NTHMO
      MP=J+NTHMO*(I-1)
      WRITE(7,426) MP
      WRITE(7,427) 1,S(I),X(I,J),Y(I,J),TTH(J)
      WRITE(7,427)2,S(I+1),X(I+1,J),Y(I+1,J),TTH(J)
      WRITE(7,427)3,S(I+1),X(I+1,J+1),Y(I+1,J+1),TTH(J+1)
      WRITE(7,427)4,S(I),X(I,J+1),Y(I,J+1),TTH(J+1)
      WRITE(7,427)5,SE(MP),XE(MP),YE(MP),THE(MP)
  203 CONTINUE
    1 FORMAT(12A6)
    2 FORMAT(2I10,F10.5)
    3 FORMAT(7F10.5)
    5 FORMAT(2F10.5,2I10)
    8 FORMAT(1H1)
   70 FORMAT(3A10)
  207 CONTINUE
  400 FORMAT(30H DO YOU WISH TO ENTER NEW BL?   )
  401 FORMAT(1F10.5)
  402 FORMAT(20H BL=      1F10.5          )
  403 FORMAT(38H   HB         THMB      BETB  3F10.5     )
  404 FORMAT(15H NTH=    1I5      )
  405 FORMAT(2I5,1F10.5)
  406 FORMAT(3F10.5)
  407 FORMAT(24H RE=          1F20.1       )
  408 FORMAT(4H RE=)
  409 FORMAT(24H NSTAT=          1I5          )
  410 FORMAT(30H     S1        S2       7F10.5   )
  411 FORMAT(25H TTH=        1F10.5            )
  412 FORMAT(1I1)
  413 FORMAT(25H END OF FILE REACHED       )
  414 FORMAT(38H NINPT DENOTES THE TYPE OF INPUT           )
  415 FORMAT(23H NINPT=1  CARD DATA      )
  416 FORMAT(23H       2  REMOTE         )
  417 FORMAT(23H       3  STORED DATA    )
```

```
  418 FORMAT(//15H NINPT    1I1             )
  419 FORMAT(43H PROVIDE READ STATEMENTS FOR CARDS AT 301      )
  420 FORMAT(27H IF YES TYPE 1 NO TYPE 0    )
  421 FORMAT(15H BL=    1F10.5   )
  422 FORMAT(20H START=     1F10.5    )
  423 FORMAT(20H SFINIS=     1F10.5    )
  424 FORMAT(20H NSTAT=      1I3       )
  425 FORMAT(20H NTH=        1I3        )
  426 FORMAT(1I5)
  427 FORMAT(1I5,4F10.5)
  428 FORMAT(12F5.3)
  429 FORMAT(1I3)
      END
      SUBROUTINE CORD(N,ST,TH,X,Y,KPK)
      COMMON /A4/ AN(10),ANP(10),ANPP(10)
      CALL BODY(ST,N,KPK)
      X=0.0
      Y=0.0
      DO 23 J=1,N
      C1=2*J
      C4=3.-C1
      CS4=COS(C4*TH)
      SS4=SIN(C4*TH)
      X=X+AN(J)*CS4
   23 Y=Y+AN(J)*SS4
      END
      SUBROUTINE BODY(ST,N,M)
      COMMON /A2/ BN(25,10)
      COMMON /A4/ AN(10),ANP(10),ANPP(10)
      COMMON /C1/ SP(50)
      M2=2*M-1
      M3=2*M-4
      M4=M2+1
      SP(1)=1.0
      DO 11 L=2,M2
   11 SP(L)=SP(L-1)*ST
      DO 22 L=M4,50
   22 SP(L)=0.0
      DO 12 J=1,N
      AN(J)=0.0
      ANP(J)=0.0
      ANPP(J)=0.0
      DO 13 L=1,M
   13 AN(J)=AN(J)+BN(L,J)*SP(L)
      DO 14 L=2,M
      C=L-1
   14 ANP(J)=ANP(J)+BN(L,J)*SP(L-1)*C
      DO 15 L=3,M
      C=(L-2)*(L-1)
   15 ANPP(J)=ANPP(J)+BN(L,J)*SP(L-2)*C
   12 CONTINUE
      RETURN
       END
```

```
#FILE (CHXL)DOUBDD ON DTNSRDC
FILE  6(KIND=PRINTER,MAXRECSIZE=132)
FILE  7(TITLE="CBL3DOPT",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
FILE  8(TITLE="DOUBDOPT",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
FILE  9(KIND=REMOTE,MAXRECSIZE=22)
FILE  11(TITLE="DBLTAPE",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
 $RESET FREE
C   PROGRAM DOUBBD
      IMPLICIT REAL*8 (A-H,$,O-Z)
      DIMENSION VPP(250,250 ),T(3500),TP(1500),XINTER(250
     1,10,3),SIG(500 ),INDEX(500 ,3),SY(2)
      DIMENSION NINTER(500 )
      DIMENSION IP(500 )
      DIMENSION VTEMP(250 ,12,10)
      DIMENSION TITLE(12)
      DIMENSION TTH(50)
      DIMENSION MP(50,50)
      COMMON /C1/ SE(250),XE(250),YE(250),THE(250)
      COMMON /C2/ S(50),X(50,50),Y(50,50)
      COMMON /C3/ NSTAT,NTH,NSMO,NTHMO
      READ (7,401) (TITLE(J),J=1,12)
      READ (7,402) N,KPK,BL
      READ(7,405) START,SFINIS,NSTAT,NTH
      NSMO=NSTAT -1
      NTHMO=NTH-1
      DO 200 I=1,NSMO
      DO 200 J=1,NTHMO
      READ (7,426) MP(I,J)
      LP=MP(I,J)
      READ (7,427) M1,S(I),X(I,J),Y(I,J),TTH(J)
      READ (7,427)M2,S(I+1),X(I+1,J),Y(I+1,J),TTH(J)
      READ (7,427)M3,S(I+1),X(I+1,J+1),Y(I+1,J+1),TTH(J+1)
      READ (7,427)M4,S(I),X(I,J+1),Y(I,J+1),TTH(J+1)
      READ (7,427)M5,SE(LP),XE(LP),YE(LP),THE(LP)
      WRITE(11,426) MP(I,J)
      WRITE(11,427) M1,S(I),X(I,J),Y(I,J),TTH(J)
      WRITE(11,427)M2,S(I+1),X(I+1,J),Y(I+1,J),TTH(J)
      WRITE(11,427)M3,S(I+1),X(I+1,J+1),Y(I+1,J+1),TTH(J+1)
      WRITE(11,427)M4,S(I),X(I,J+1),Y(I,J+1),TTH(J+1)
      WRITE(11,427)M5,SE(LP),XE(LP),YE(LP),THE(LP)
  200 CONTINUE
      NP=(NSTAT-1)*(NTH-1)
      IF(NP.GT.250) STOP
```

```
      WRITE(8,402) NP
      WRITE(8,406) NSMO,NTHMO
      DO 201 I=1,250
      DO 201 J=1,10
      DO 201 K=1,3
  201 XINTER(I,J,K)=0.0D00
      CALL XPAPS(NP,VPP,T,TP,XINTER,SIG,SY,VTEMP,NINTER,INDEX)
      SY(1)=1.0
      SY(2)=-1.0
      FPI=12.5664D00
      DO 10 I=1,NP
      L=(I-1)*3+1
      K=(I-1)*7+1
      TP(L)=T(K)
      TP(L+1)=T(K+1)
      TP(L+2)=T(K+2)
      SIG(I)=-TP(L)*12.5664D00
      DO 10 J=1,NP
      VPP(I,J)=0.0
   10 CONTINUE
      NUMB=0
      DO 20 I=1,NP
      NUMB=MAX0(NINTER(I),NUMB)
   20 CONTINUE
      DO 21 ITER=1,10
      DO 21 I=1,NP
      INDEX(I,1)=NINTER(I)-ITER
      VTEMP(I,12,ITER)=DFLOAT(MIN0(INDEX(I,1),0))+1.0D00
      VTEMP(I,12,ITER)=DMAX1(VTEMP(I,12,ITER),0.0D00)
   21 CONTINUE
      WRITE(6,1001)(VTEMP(I,12,1),I=1,NP)
      WRITE(6,1001)(VTEMP(I,12,3),I=1,NP)
      DO 1 I=1,NP
      DO 23 J=1,NP
      VTEMP(J,11,1)=0.0D00
   23 CONTINUE
      VTEMP(I,11,1)=1.0D10
      L=(I-1)*3+1
      DO 4 KY=1,2
   24 DO 2 ITER=1,NUMB
   22 DO 3 J=1,NP
      K=(J-1)*7+1
      VTEMP(J,KY,ITER)=TP(L)-XINTER(J,ITER,1)
      VTEMP(J,KY,ITER)=VTEMP(J,KY,ITER)+VTEMP(J,11,1)
      VTEMP(J,KY+2,ITER)=TP(L+1)-XINTER(J,ITER,2)*SY(KY)
      VTEMP(J,KY+4,ITER)=TP(L+2)-XINTER(J,ITER,3)
      VTEMP(J,KY+6,ITER)=VTEMP(J,KY,ITER)**2+VTEMP(J,KY+2,ITER)**2
     1+VTEMP(J,KY+4,ITER)**2
      VTEMP(J,KY+8,ITER)=DSQRT(VTEMP(J,KY+6,ITER))
      VTEMP(J,KY+8,ITER)=VTEMP(J,KY+8,ITER)*VTEMP(J,KY+6,ITER)
      VPP(I,J)=VPP(I,J)+(VTEMP(J,KY,ITER)*T(K+3)+VTEMP(J,KY+2,ITER)+SY(K
```

```
     1Y)*T(K+4)+VTEMP(J,KY+4,ITER)*T(K+5))*T(K+6)/VTEMP(J,KY+8,ITER)
     2*VTEMP(J,12,ITER)
    3 CONTINUE
    2 CONTINUE
    4 CONTINUE
      DO 17 ITER=1,NUMB
      DO 18 KY=1,2
      DO 7 J=1,NP
       K=(J-1)*7+1
       VTEMP(J,KY+4,ITER)=TP(L+2)+XINTER(J,ITER,3)
       VTEMP(J,KY+6,ITER)=VTEMP(J,KY,ITER)**2+VTEMP(J,KY+2,ITER)**2
      1+VTEMP(J,KY+4,ITER)**2
       VTEMP(J,KY+8,ITER)=DSQRT(VTEMP(J,KY+6,ITER))
       VTEMP(J,KY+8,ITER)=VTEMP(J,KY+8,ITER)*VTEMP(J,KY+6,ITER)
       VPP(I,J)=VPP(I,J)+(VTEMP(J,KY,ITER)*T(K+3)+VTEMP(J,KY+2,ITER)*T(K+
      14)*SY(KY)+VTEMP(J,KY+4,ITER)*(-T(K+5)))/VTEMP(J,KY+8,ITER)*T(K+6)
      2*VTEMP(J,12,ITER)
     7 CONTINUE
    18 CONTINUE
    17 CONTINUE
     1 CONTINUE
       DO 12 I=1,NP
       VPP(I,I)=0.0
       VTEMP(I,11,1)=0.0D00
    12 CONTINUE
       DO 8 I=1,NP
       DO 6 J=1,NP
       VPP(I,J)=VPP(I,J)/DFLOAT(NINTER(J))
     6 CONTINUE
     8 CONTINUE
       DO 15 I=1,NP
       SUM=0.0D00
       DO 14 J=1,NP
       SUM=SUM+VPP(I,J)
    14 CONTINUE
       VTEMP(I,11,1)=SUM
    15 CONTINUE
       DO 9 I=1,NP
       VPP(I,I)=-FPI-VTEMP(I,11,1)
     9 CONTINUE
       WRITE(6,1001)(VPP(I,I),I=1,NP)
       WRITE(6,1001)(VPP(I,1),I=1,NP,5)
           WRITE (11,1002) (VPP(I,I),I=1,NP)
       WRITE(6,1003)(NINTER(I),I=1,NP)
       WRITE(6,1003) NUMB
       WRITE(6,1001)(((XINTER(I,J,K),K=1,3),J=1,3),VTEMP(I,11,1),I=1,NP)
       CALL DECOMP(NP,NP,VPP,IP)
       CALL SOLVE(NP,NP,VPP,SIG,IP)
       WRITE(6,1000)
       DO 5 I=1,NP
       L=(I-1)*3
```

```
          WRITE(8,400) (-TP(L+1),TP(L+2),TP(L+3),-SIG(I))
    5 WRITE(6,1001) (TP(L+1),TP(L+2),TP(L+3),SIG(I))
         WRITE (11,1002) ((T(7*I-7+J),J=1,7),SIG(I),I=1,NP)
      STOP
 1002 FORMAT(8D10.4)
 1000 FORMAT('1',20X,'POTENTIAL')
 1001 FORMAT(1H ,10D12.5)
  400 FORMAT(10E12.5)
  401 FORMAT(12A6)
  402 FORMAT(2I10,1F10.5)
  405 FORMAT(2F10.5,2I10)
  406 FORMAT(2I10)
  426 FORMAT(1I5)
  427 FORMAT(1I5,4F10.5)
 1003 FORMAT(8I10)
      END
       SUBROUTINE MATINS(A,NR,N1,B,NC,M1,DETERM,ID,INDEX)
C       PIVOT METHOD                                                    M
C        MATRIX  INVERSION WITH ACCOMPANYING SOLUTION OF SIMUL. EQ.
C     PIVOT METHOD
C     FORTRAN IV SINGLE PRECISION WITH ADJUSTABLE DIMENSION
C     FEBRUARY 1966   S GOOD  DAVID TAYLOR MODEL BASIN   AM MAT4
C          WHERE CALLING PROGRAM MUST INCLUDE
C              DIMENSION A(      ), B(      ), INDEX(     )
C           N     IS THE ORDER OF A
C           M     IS THE NUMBER OF COLUMN VECTORS IN B(MAY BE 0)
C          DETERM WILL CONTAIN DETERMINANT ON EXIT
C           ID    WILL BE SET BY ROUTINE TO 2 IF MATRIX A IS SINGULAR
C                 1 IF INVERSION WAS SUCCESSFUL
C           A     THE INPUT MATRIX WILL BE REPLACED BY A INVERSEE
C           B     THE COLUMN VECTORS WILL BE REPLACED BY CORRESPONDING
C                 SOLUTION VECTORS
C           INDEX WORKING STORAGE ARRAY
C          IF IT IS DESIRED TO SCALE THE DETERMINANT CARD      MAY BE
C          DELETED AND DETERM PRESET BEFORE ENTERING THE ROUTINE
C
      IMPLICIT REAL*8 (A-H,$,O-Z)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T)
      DIMENSION A(NR ,NR ),B(NR ,NC ),INDEX(NR ,3)
      DIMENSION VT(500),IL1(500),SWAP(500)
C
C      INITIALIZATION
C
       N=N1
      M=M1
       DETERM = 1.0
       DO 20 J=1,N
   20 INDEX(J,3) = 0
       DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
```

```
C
      AMAX = 0.0
      DO 105 J=1,N
      IF(INDEX(J,3)-1) 60, 105, 60
   60 DO 100 K=1,N
      IF(INDEX(K,3)-1) 80, 100, 715
   80 IF (      AMAX -DABS (A(J,K))) 85, 100, 100
   85 IROW=J
      ICOLUM =K
      AMAX = DABS (A(J,K))
  100  CONTINUE
  105  CONTINUE
      INDEX(ICOLUM,3) = INDEX(ICOLUM,3) +1
      INDEX(I,1)=IROW
C
      INDEX(I,2)=ICOLUM
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
      IF (IROW-ICOLUM) 140, 310, 140
  140 DETERM=-DETERM
      DO 201 L=1,N
      SWAP(L)=A(IROW,L)
  201 CONTINUE
      DO 202 L=1,N
      A(IROW,L)=A(ICOLUM,L)
  202 CONTINUE
      DO 200 L=1,N
  200 A(ICOLUM,L)=SWAP(L)
      IF(M) 310, 310, 210
  210 DO 251 L=1, M
      SWAP(L)=B(IROW,L)
  251 CONTINUE
      DO 252 L=1,M
      B(IROW,L)=B(ICOLUM,L)
  252 CONTINUE
      DO 250 L=1,M
  250 B(ICOLUM,L)=SWAP(L)
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  310 PIVOT    =A(ICOLUM,ICOLUM)
C     DETERM=DETERM*PIVOT
  330 A(ICOLUM,ICOLUM)=1.0
      DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
      IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
C
C     REDUCE NON-PIVOT ROWS
C
```

```
  380 DO 550 L1=1,N
      IL1(L1)=L1
      IF(IL1(L1)-ICOLUM) 400, 550, 400
  400 VT(L1)=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.0
      DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*VT(L1)
      IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*VT(L1)
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
      DO 710 I=1,N
      L=N+1-I
      IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
      JCOLUM=INDEX(L,2)
      DO 706 K=1,N
      SWAP(K)=A(K,JROW)
  706 CONTINUE
      DO 707 K=1,N
      A(K,JROW)=A(K,JCOLUM)
  707 CONTINUE
      DO 705 K=1,N
      A(K,JCOLUM)=SWAP(K)
  705 CONTINUE
  710 CONTINUE
      DO 730 K = 1,N
      IF(INDEX(K,3) -1) 715,720,715
 720   CONTINUE
 730   CONTINUE
       ID = 1
 810  RETURN
 715  ID = 2
       GO TO 810
       END
      SUBROUTINE SOLVE(N,NDIM,A,B,IP)
      IMPLICIT REAL*8 (A-H,$,O-Z)
      DIMENSION A(NDIM ,NDIM ),B(NDIM ),IP(NDIM )
      IF(N.EQ.1) GO TO 9
      NM1=N-1
      DO 7 K=1,NM1
      KP1=K+1
      M=IP(K)
      T=B(M)
      B(M)=B(K)
      B(K)=T
      DO 7 I=KP1,N
    7 B(I)=B(I)+A(I,K)*T
```

```
      DO 8 KB=1,NM1
      KM1=N-KB
      K=KM1+1
      B(K)=B(K)/A(K,K)
      T=-B(K)
      DO 8 I=1,KM1
    8 B(I)=B(I) +A(I,K)*T
    9 B(1)=B(1)/A(1,1)
      RETURN
      END
      SUBROUTINE XPAPS(NP,VPP,T,TP,XINTER,SIG,SY,VTEMP,NINTER,INDEX)
      IMPLICIT REAL*8 (A-H,$,O-Z)
      COMMON /C1/ SE(250),XE(250),YE(250),THE(250)
      COMMON /C2/ S(50),X(50,50),Y(50,50)
      COMMON /C3/ NSTAT,NTH,NSMO,NTHMO
      DIMENSION VPP(NP ,NP ),T(3500),TP(1500),XINTER(NP ,
     110,3),SIG(NP ),INDEX(NP ,3),SY(2)
      DIMENSION NINTER(NP )
      DIMENSION VTEMP(NP ,12,10)
      WRITE(6,1000) NP
      L=0
      DO 200 I=1,NSMO
      DO 200 J=1,NTHMO
      L=L+1
      VPP(L,1)=-S(I)
      VPP(L,2)=X(I,J)
      VPP(L,3)=Y(I,J)
      VPP(L,4)=-S(I+1)
      VPP(L,5)=X(I+1,J)
      VPP(L,6)=Y(I+1,J)
      VPP(L,7)=-S(I+1)
      VPP(L,8)=X(I+1,J+1)
      VPP(L,9)=Y(I+1,J+1)
      VPP(L,10)=-S(I)
      VPP(L,11)=X(I,J+1)
      VPP(L,12)=Y(I,J+1)
      VPP(L,13)=-SE(L)
      VPP(L,14)=XE(L)
      VPP(L,15)=YE(L)
      WRITE(11,402)I,J,L
      WRITE(11,403)(VPP(L,IP),IP=1,15)
  402 FORMAT(3I5)
  403 FORMAT(5F10.5)
  200 CONTINUE
      DO 1 I=1,NP
      L=(I-1)*3
      DO 1 J=1,3
      TP(L+J)=VPP(I,12+J)
    1 CONTINUE
      DO 11 I=1,NP
      VTEMP(I,1,1)=VPP(I,10)-VPP(I,4)
```

```
      VTEMP(I,2,1)=VPP(I,1)-VPP(I,7)
      VTEMP(I,3,1)=VPP(I,11)-VPP(I,5)
      VTEMP(I,4,1)=VPP(I,2)-VPP(I,8)
      VTEMP(I,5,1)=VPP(I,12)-VPP(I,6)
      VTEMP(I,6,1)=VPP(I,3)-VPP(I,9)
      VTEMP(I,4,2)= VTEMP(I,4,1)*VTEMP(I,5,1)-VTEMP(I,3,1)*VTEMP(I,6,1)
      VTEMP(I,5,2)= VTEMP(I,1,1)*VTEMP(I,6,1)-VTEMP(I,2,1)*VTEMP(I,5,1)
      VTEMP(I,6,2)= VTEMP(I,2,1)*VTEMP(I,3,1)-VTEMP(I,1,1)*VTEMP(I,4,1)
      VTEMP(I,1,10)=DSQRT(VTEMP(I,4,2)**2+VTEMP(I,5,2)**2+VTEMP(I,6,2)**
     12)
      WRITE(11,400)I,(VTEMP(I,J,1),J=1,6),(VTEMP(I,J,2),J=4,6)
      WRITE(11,401) VTEMP(I,1,10)
  400 FORMAT(1I5,/6F10.5,/3F10.5)
  401 FORMAT(1E15.8)
      V=VTEMP(I,1,10)
      TEST=DABS(V)
      IF(TEST .LT.1.0E- 8)GOTO 11
      VTEMP(I,4,2)=VTEMP(I,4,2)/VTEMP(I,1,10)
      VTEMP(I,5,2)=VTEMP(I,5,2)/VTEMP(I,1,10)
      VTEMP(I,6,2)=VTEMP(I,6,2)/VTEMP(I,1,10)
      VTEMP(I,7,2)=.5*VTEMP(I,1,10)
C        G3  COMPUTE COVTEMP(I,1,10)NEVTEMP(I,1,10) PTS. (VTEMP(I,1,1),VT
      VTEMP(I,1,1)=(VPP(I,1)+VPP(I,4)+VPP(I,7)+VPP(I,10))*.25
      VTEMP(I,2,1)=(VPP(I,2)+VPP(I,5)+VPP(I,8)+VPP(I,11))*.25
      VTEMP(I,3,1)=(VPP(I,3)+VPP(I,6)+VPP(I,9)+VPP(I,12))*.25
      VTEMP(I,4,1)=VTEMP(I,4,2)*(VTEMP(I,1,1)-VPP(I,4))+VTEMP(I,5,2)*(
     1VTEMP(I,2,1)-VPP(I,5))+VTEMP(I,6,2)*(VTEMP(I,3,1)-VPP(I,6))
      VTEMP(I,1,3)=VTEMP(I,4,2)*VTEMP(I,4,1)
      VTEMP(I,2,3)=VTEMP(I,5,2)*VTEMP(I,4,1)
      VTEMP(I,3,3)=VTEMP(I,6,2)*VTEMP(I,4,1)
      VTEMP(I,1,1)=VPP(I,4)+VTEMP(I,1,3)
      VTEMP(I,2,1)=VPP(I,5)+VTEMP(I,2,3)
      VTEMP(I,3,1)=VPP(I,6)+VTEMP(I,3,3)
      VTEMP(I,4,1)=VPP(I,7)-VTEMP(I,1,3)
      VTEMP(I,5,1)=VPP(I,8)-VTEMP(I,2,3)
      VTEMP(I,6,1)=VPP(I,9)-VTEMP(I,3,3)
      VTEMP(I,7,1)=VPP(I,10)+VTEMP(I,1,3)
      VTEMP(I,8,1)=VPP(I,11)+VTEMP(I,2,3)
      VTEMP(I,9,1)=VPP(I,12)+VTEMP(I,3,3)
      VTEMP(I,10,1)=VPP(I,1)-VTEMP(I,1,3)
      VTEMP(I,11,1)=VPP(I,2)-VTEMP(I,2,3)
      VTEMP(I,12,1)=VPP(I,3)-VTEMP(I,3,3)
C        G4  COMPUTE CENTVTEMP(I,1,10)OID (VTEMP(I,1,2),VTEMP(I,2,2),VTEM
  345 VTEMP(I,1,3)=VTEMP(I,4,1)-VTEMP(I,1,1)
      VTEMP(I,2,3)=VTEMP(I,10,1)-VTEMP(I,1,1)
      VTEMP(I,3,3)=VTEMP(I,5,1)-VTEMP(I,2,1)
      VTEMP(I,4,3)=VTEMP(I,11,1)-VTEMP(I,2,1)
      VTEMP(I,5,3)=VTEMP(I,6,1)-VTEMP(I,3,1)
      VTEMP(I,6,3)=VTEMP(I,12,1)-VTEMP(I,3,1)
      VTEMP(I,1,4)=VTEMP(I,3,3)*VTEMP(I,6,3)-VTEMP(I,4,3)*VTEMP(I,5,3)
      VTEMP(I,2,4)=VTEMP(I,5,3)*VTEMP(I,2,3)-VTEMP(I,1,3)*VTEMP(I,6,3)
```

```
      VTEMP(I,3,4)=VTEMP(I,1,3)*VTEMP(I,4,3)-VTEMP(I,2,3)*VTEMP(I,3,3)
      VTEMP(I,7,3)=DSQRT(VTEMP(I,1,4)**2+VTEMP(I,2,4)**2+VTEMP(I,3,4)**2
     1)
  348 VTEMP(I,3,2)=(VTEMP(I,6,1)+VTEMP(I,12,1)+(VTEMP(I,7,3)*VTEMP(I,3,1
     1)+(VTEMP(I,1,10)-VTEMP(I,7,3))*VTEMP(I,9,1))/VTEMP(I,1,10))/3.
      VTEMP(I,2,2)=(VTEMP(I,5,1)+VTEMP(I,11,1)+(VTEMP(I,7,3)*VTEMP(I,2,1
     1)+(VTEMP(I,1,10)-VTEMP(I,7,3))*VTEMP(I,8,1))/VTEMP(I,1,10))/3.
      VTEMP(I,1,2)=(VTEMP(I,4,1)+VTEMP(I,10,1)+(VTEMP(I,7,3)*VTEMP(I,1,1
     1)+(VTEMP(I,1,10)-VTEMP(I,7,3))*VTEMP(I,7,1))/VTEMP(I,1,10))/3.
   11 CONTINUE
C        H3  STOVTEMP(I,1,10)E CENTVTEMP(I,1,10)OID AND AVTEMP(I,1,10)EA
      DO 10 I=1,NP
      K=(I-1)*7
  950 T(K+1)=VTEMP(I,1,2)
      T(K+2)=VTEMP(I,2,2)
      T(K+3)=VTEMP(I,3,2)
C        H4  STOVTEMP(I,1,10)E NOVTEMP(I,1,10)MAL VECTOVTEMP(I,1,10)
      T(K+4)=VTEMP(I,4,2)
      T(K+5)=VTEMP(I,5,2)
      T(K+6)=VTEMP(I,6,2)
      T(K+7)=VTEMP(I,7,2)
   10 CONTINUE
      DO 70 I=1,NP
      VTEMP(I,1,5)=(VPP(I,4)+VPP(I,7))*0.5
      VTEMP(I,2,5)=(VPP(I,5)+VPP(I,8))*0.5
      VTEMP(I,3,5)=(VPP(I,6)+VPP(I,9))*0.5
      VTEMP(I,1,6)=(VPP(I,1)+VPP(I,10))*0.5
      VTEMP(I,2,6)=(VPP(I,2)+VPP(I,11))*0.5
      VTEMP(I,3,6)=(VPP(I,3)+VPP(I,12))*0.5
      NINTER(I)=(VTEMP(I,1,6)-VTEMP(I,1,5))**2/(VTEMP(I,7,2)*1.5D00)
      NINTER(I)=MAX0(NINTER(I),1)
      NINTER(I)=MIN0(NINTER(I),8)
      NUMB=NINTER(I)
      VTEMP(I,1,6)=(VTEMP(I,1,6)-VTEMP(I,1,5))/NINTER(I)
      VTEMP(I,2,6)=(VTEMP(I,2,6)-VTEMP(I,2,5))/NINTER(I)
      VTEMP(I,3,6)=(VTEMP(I,3,6)-VTEMP(I,3,5))/NINTER(I)
      DO 999 NIX=1,NUMB
      DNIX=DFLOAT(NIX)-0.5D00
      XINTER(I,NIX,1)=VTEMP(I,1,5)+VTEMP(I,1,6)*DNIX
      XINTER(I,NIX,2)=VTEMP(I,2,5)+VTEMP(I,2,6)*DNIX
      XINTER(I,NIX,3)=VTEMP(I,3,5)+VTEMP(I,3,6)*DNIX
  999 CONTINUE
      IF(NINTER(I).NE.1) GO TO 998
      DO 997 NIX=1,3
      XINTER(I,1,NIX)=VTEMP(I,NIX,2)
  997 CONTINUE
  998 CONTINUE
C     WX(I)=VPP(I,1)-VPP(I,4)
   70 CONTINUE
C     IF(LIN.LT.60) GO TO 80
      WRITE (6,7)
```

```
      WRITE (6,8)
      LIN=0
C  80 LIN=LIN+1
      DO 55 I=1,NP
      M=(I-1)*7+1
   55 WRITE (6,6) T(M),T(M+1),T(M+2),T(M+3),T(M+4),T(M+5),T(M+6)
    4 CONTINUE
      NQE=NP
      NQ7=NQE*7
      WRITE(6,28) NP
    6 FORMAT (10D12.5)
    7 FORMAT(38H1 INPUT DATA TO POTENTIAL FLOW PROGRAM)
    8 FORMAT(4H0   ,2HXP,10X,2HYP,10X,2HZP,10X,2HXN,10X,2HYN,10X,2HZN
     1,10X,2HAQ)
   24 FORMAT (1H ,I1,19H PLANES OF SYMMETRY)
   28 FORMAT (33HOTOTAL NUMBER OF QUADRILATE =    ,I5)
 1000 FORMAT(10I5)
 1001 FORMAT(16F5.1)
      RETURN
      END
      SUBROUTINE DECOMP(N,NDIM,A,IP)
      IMPLICIT REAL*8 (A-H,$,O-Z)
      DIMENSION  A(NDIM,NDIM),IP(NDIM)
      IP(N)=1
      DO 6 K=1,N
      IF(K.EQ.N) GO TO 5
      KP1=K+1
      M=K
      DO 1 I=KP1,N
      IF(DABS(A(I,K)).GT.DABS(A(M,K))) M=I
    1 CONTINUE
      IP(K)=M
      IF(M.NE.K) IP(N)=-IP(N)
      T=A(M,K)
      A(M,K)=A(K,K)
      A(K,K)=T
      IF(T.EQ.0.) GO TO 5
      DO 2 I=KP1,N
    2 A(I,K)=-A(I,K)/T
      DO 4 J=KP1,N
      T=A(M,J)
      A(M,J)=A(K,J)
      A(K,J)=T
      IF(T.EQ.0.) GO TO 4
      DO 3 I=KP1,N
    3 A(I,J)=A(I,J)+A(I,K)*T
    4 CONTINUE
    5 IF(A(K,K).EQ.0.) IP(N)=0
    6 CONTINUE
      RETURN
      END
```

```
100 FILE  5(KIND=REMOTE,MAXRECSIZE=22)
110 FILE  6(KIND=PRINTER,MAXRECSIZE=132)
120 FILE  7(TITLE="TBLIPCOPT",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
130 FILE  8(TITLE="TBL3DOPT",KIND=DISK,FILETYPE=7)
140 FILE  9(KIND=REMOTE,MAXRECSIZE=22)
150 FILE  10(TITLE="TBL3DINP",KIND=DISK,FILETYPE=7)
160 FILE  11(TITLE="TBLSAVE",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
165 FILE  12(TITLE="THREEDPHI",KIND=DISK,FILETYPE=7)
170  $RESET FREE
1010 C      PROGRAM TBLIPC
1030 C      PROGRAM COMPUTES DERIVATIVES OF VELOCITIES ALONG THE ORTHOGONAL
1040 C      COORDINATES.
1050        DIMENSION TITLE(12),TITL(3)
1060        DIMENSION X(3,120),Y(3,120),TH(3,120),UP(3,120),UT(3,120),S(3,120)
1070        DIMENSION XKP(3,120),XKT(3,120),H(3,120),F(3,120),G(3,120)
1080        DIMENSION JJ(3),SS(3)
1090        COMMON/SPLIN/SP(30,60),SPHI(30,60),SPHP(30,60),SPHPP(30,60),
1100       $ DTP(30,60)
1110        COMMON/VELOC/UP,UT,S,TH
1120        READ (8,401) (TITLE(J),J=1,12)
1130        WRITE(6,401) (TITLE(J),J=1,12)
1140        WRITE(7,401) (TITLE(J),J=1,12)
1145        WRITE(9,401) (TITLE(J),J=1,12)
1180        READ (8,402 ) N,KPK,BL
1190        WRITE(6,402 ) N,KPK,BL
1200        WRITE(7,402 ) N,KPK,BL
1205        WRITE(9,402) N,KPK,BL
1210        READ (8,405) START,SFINS,NSTAT,NTH
1220        WRITE(6,405) START,SFINS,NSTAT,NTH
1230        WRITE(7,405) START,SFINS,NSTAT,NTH
1235        WRITE(9,405) START,SFINS,NSTAT,NTH
1240        PIOT=1.57079682679
1250        EN=NTH-1
1260        MTH=NTH-1
1270        DT=PIOT/EN*(-1.0)
1280        DTI=1.0/DT
1290        WRITE(9,408)NTH,DT,DTI
1300        DTF=0.5*DTI
1310        J=1
1320        K=2
1330        L=3
1331        WRITE(9,406)
1332        WRITE(9,407)
1335        READ (5,404) NS
1338        IF(NS.EQ.0) GO TO 300
```

```
      READ (12,403)NS,NT
      WRITE(6,403)NS,NT
      CALL SPOTEN(NS,NT,DTH,NTH)
      TEST=SP(1,1)
  300 CONTINUE
      DO 205 I=1,2
  301 CONTINUE
      READ (8,427) JJ(I),SS(I)
      WRITE(6.427) JJ(I),SS(I)
      DO200 IT=1,NTH
      READ (8,408) NDUM,X(I,IT),Y(I,IT),TH(I,IT),U,V
      WRITE(6,408) NDUM,X(I,IT),Y(I,IT),TH(1,IT),U,V
      READ (8,408) NDUM,W,UP(I,IT),UT(I,IT),UN,S(I,IT)
      WRITE(6,408) NDUM,W,UP(I,IT),UT(I,IT),UN,S(I,IT)
      READ (8,408) NDUM,XKP(I,IT),XKT(I,IT),H(I,IT),F(I,IT),G(I,IT)
      WRITE(6,408) NDUM,XKP(I,IT),XKT(I,IT),H(I,IT),F(I,CT),G(I,IT)
  200 CONTINUE
      NSTAT=NSTAT-1
      IF(TEST.GE.SS(1)) GOTO 301
      NSTAT=NSTAT+1
  205 CONTINUE
      IF(NS.EQ.0) GO TO 302
      CALL EVALVO(NS,DTH,NTH,1)
      CALL EVALVO(NS,DTH,NTH,2)
  302 CONTINUE
       HI=1.0/H(2,1)
      DTIG=DTI/G(1,1)
      DUPDLP=(UP(2,1)-UP(1,1))/(S(2,1)-S(1,1))
       DUPDLP=HI*DUPDLP
      DUPDLT=DTIG*(       UP(1,2)-UP(1,1))
      DUTDLP=(UT(2,1)-UT(1,1))/(S(2,1)-S(1,1))
      DUTDLP=HI*DUTDLP
      DUTDLT=DTIG*(UT(1,2)-UT(1,1))
      WRITE(7,427) JJ(1),SS(1)
      WRITE(6,400) S(1,1)
      WRITE(6,408) 1,X(1.1),Y(1,1),TH(1,1),UP(1,1),UT(1,1)
      WRITE(7,408) 1,X(1,1),Y(1,1),TH(1,1),UP(1,1),UT(1,1)
      WRITE(6,408) 2,S(1,1),DUPDLP,DUPDLT,DUTDLP,DUTDLT
      WRITE(7,408) 2,S(1,1),DUPDLP,DUPDLT,DUTDLP,DUTDLT
      WRITE(6,408) 3,XKP(1,1),XKT(1,1),H(1,1),F(1,1),G(1,1)
      WRITE(7,408) 3,XKP(1,1),XKT(1,1),H(1,1),F(1,1),G(1,1)
      DO 201 IT=2,MTH
      DS=1.0/(S(2,IT)-S(1,IT))
      DS=DS/H(2,IT)
      DUPDLP=DS*(UP(2,IT)-UP(1,IT))
```

```
      DUTDLP=DS*(UT(2,IT)-UT(1,IT))
      DTFG=DTF/G(1,IT)
      DUPDLT=DTFG*(UP(1,IT+1)-UP(1,IT-1))
      DUTDLT=DTFG*(UT(1,IT+1)-UT(1,IT-1))
      WRITE(6,408) 1,X(1,IT),Y(1,IT),TH(1,IT),UP(1,IT),UT(1,IT)
      WRITE(7,408) 1,X(1,IT),Y(1,IT),TH(1,IT),UP(1,IT),UT(1,IT)
      WRITE(6,408) 2,S(1,IT),DUPDLP,DUPDLT,DUTDLP,DUTDLT
      WRITE(7,408) 2,S(1,IT),DUPDLP,DUPDLT,DUTDLP,DUTDLT
      WRITE(6,408) 3,XKP(1,IT),XKT(1,IT),H(1,IT),F(1,IT),G(1,IT)
      WRITE(7,408) 3,XKP(1,IT),XKT(1,IT),H(1,IT),F(1,IT),G(1,IT)
  201 CONTINUE
      DS=1.0/(S(2,NTH)-S(1,NTH))
      DS=DS/H(2,NTH)
      DUPDLP=DS*(UP(2,NTH)-UP(1,NTH))
      DUTDLP=DS*(UT(2,NTH)-UT(1,NTH))
      DTIG=DTI/G(1,NTH)
      DUPDLT=DTIG*(UP(1,NTH)-UP(1,MTH))
      DUTDLT=DTIG*(UT(1,NTH)-UT(1,MTH))
      WRITE(6,408) 1,X(1,NTH),Y(1,NTH),TH(1,NTH),UP(1,NTH),UT(1,NTH)
      WRITE(7,408) 1,X(1,NTH),Y(1,NTH),TH(1,NTH),UP(1,NTH),UT(1,NTH)
      WRITE(6,408) 2,S(1,NTH),DUPDLP,DUPDLT,DUTDLP,DUTDLT
      WRITE(7,408) 2,S(1,NTH),DUPDLP,DUPDLT,DUTDLP,DUTDLT
      WRITE(6,408) 3,XKP(1,NTH),XKT(1,NTH),H(1,NTH),F(1,NTH),G(1,NTH)
      WRITE(7,408) 3,XKP(1,NTH),XKT(1,NTH),H(1,NTH),F(1,NTH),G(1,NTH)
C     OUTPUT FOR FIRST PHI-STATION IS COMPLETED.
      DO 202 IS=3,NSTAT
      WRITE(7,427) JJ(K),SS(K)
      READ (8,427) JJ(L),SS(L)
      WRITE(6,427) JJ(L),SS(L)
      DO 203 IT=1,NTH
      READ (8,408) NDUM,X(L,IT),Y(L,IT),TH(L,IT),U,V
      READ (8,408) NDUM,W,UP(L,IT),UT(L,IT),UN,S(L,IT)
      READ (8,408) NDUM,XKP(L,IT),XKT(L,IT),H(L,IT),F(L,IT),G(L,IT)
  203 CONTINUE
      IF(NS.EQ.0) GO TO 303
      CALL EVALVO(NS,DTH,NTH,L)
  303 CONTINUE
      DTIG=DTI/G(K,1)
      DS=1.0/(S(L,1)-S(J,1))
      DS=DS/H(K,1)
      DUPDLP=DS*(UP(L,1)-UP(J,1))
      DUTDLP=DS*(UT(L,1)-UT(J,1))
      DUPDLT=DTIG*(UP(K,2)-UP(K,1))
```

```
      DUTDLT=DTIG*(UT(K,2)-UT(K,1))
      WRITE(6,400) S(K,1)
      WRITE(7,408) 1,X(K,1),Y(K,1),TH(K,1),UP(K,1),UT(K,1)
      WRITE(6,408) 2,S(K,1)       ,DUPDLP,DUPDLT,DUTDLP,DUTDLT
      WRITE(7,408) 2,S(K,1)       ,DUPDLP,DUPDLT,DUTDLP,DUTDLT
      WRITE(7,408) 3,XKP(K,1),XKT(K,1),H(K,1),F(K,1),G(K,1)
      DO 204 IT=2,NTH
      DS=1.0/(S(L,IT)-S(J,IT))
      DS=DS/H(K,IT)
      DTFG=DTF/G(K,IT)
      DUPDLP=DS*(UP(L,IT)-UP(J,IT))
      DUTDLP=DS*(UT(L,IT)-UT(J,IT))
      DUPDLT=DTFG*(UP(K,IT+1)-UP(K,IT-1))
      DUTDLT=DTFG*(UT(K,IT+1)-UT(K,IT-1))
      WRITE(7,408) 1,X(K,IT),Y(K,IT),TH(K,IT),UP(K,IT),UT(K,IT)
      WRITE(7,408) 2,S(K,IT),DUPDLP,DUPDLT,DUTDLP,DUTDLT
      WRITE(7,408) 3,XKP(K,IT),XKT(K,IT),H(K,IT),F(K,IT),G(K,IT)
  204 CONTINUE
      DS=1.0/(S(L,NTH)-S(J,NTH))
      DS=DS/H(K,NTH)
      DTIG=DTI/G(K,NTH)
      DUPDLP=DS*(UP(L,NTH)-UP(J,NTH)     )
      DUTDLP=DS*(UT(L,NTH)-UT(J,NTH)    )
      DUPDLT=DTIG*(UP(K,NTH)-UP(K,NTH))
      DUTDLT=DTIG*(UT(K,NTH)-UT(K,NTH))
      WRITE(7,408) 1,X(K,NTH),Y(K,NTH),TH(K,NTH),UP(K,NTH),UT(K,NTH)
      WRITE(7,408) 2,S(K,NTH),DUPDLP,DUPDLT,DUTDLP,DUTDLT
      WRITE(7,408) 3,XKP(K,NTH),XKT(K,NTH),H(K,NTH),F(K,NTH),G(K,NTH)

C     COMPLETED OUTPUT FOR K-TH PHI-STATION.

      M=J
      J=K
      K=L
  202 L=M

CC     DATA COMPLETED FOR ALL BUT LAST PHI-STATION.

  500 CONTINUE
  400 FORMAT(5X,28H W.L. STATION COORDINATE IS     ,1F7.4)
  401 FORMAT(12A6)
  402 FORMAT(2I10,1F10.5)
  403 FORMAT(2I5)
  404 FORMAT(1I1)
  405 FORMAT(2F10.5,2I10)
  406 FORMAT(55H IF THREE DIMENSIONAL POTENTIAL IS FROM MING'S PROGRAM )
  407 FORMAT(33H ENTER 1 SLENDER BODY ENTER 0        )
  408 FORMAT(1I3,5E15.7)
  427 FORMAT(1I3,1F9.6)
  470 FORMAT(3A10)
      END
```

```
      SUBROUTINE EVALVO(NS,DTH,NTH,IS)

C   COMPUTES UP AND  UT FROM SPLINE DATA
C        SP         S COORDINATES AT WHICH SPHI IS GIVEN
C        SPHI       SPLINE POTENTIAL
C        SPHP       DERIVATIVE OF SPHI IN S DIRECTION
C        SPHPP      SECOND DERIVATIVE OR  MOMENT OF SPHI
C        DTP        SPACING INCREMENTS IN S

      COMMON/SPLIN/SP(30,60),SPHI(30,60),SPHP(30,60),SPHPP(30,60),
     $ DTP(30,60)

C        UP         VELOCITY IN PHI DIRECTION
C        UT         VELOCITY  IN THATA DIRECTION
C        S          S COORDINATE AT WHICH  VELOCITIES ARE EVALUATED
C        TH         THATA COORDINATE

      COMMON/VELOC/UP,UT,S,TH
      DIMENSION UP(3,120),UT(3,120),S(3,120),TH(3,120)


C        SPS        INTERPOLATION POINTS FOR LONGITUDINAL SPLINE
C        PH         POTENTIAL AT SPS
C        FM         MOMENTS
C        DT         S INCREMENTS
C        T          INTERPOLATION POINTS THATA DIRECTION
C        SPHIS      POTENTIAL AT T
C        PHP        DERIVATIVE OF THE POTENTIAL
C        DELTA      INCREMENTS IN T
C        MY         MOMENTS

      DIMENSION SPS(120),PH(120),FM(120),DT(120),T(120),SPHIS(120)
     $,PHP(120) ,DELTA(120),MY(120)
      REAL MY
      PI=3.1415962
      TOL=1.0E-06
      MTH=2*NTH

C   COMPUTE UP

      DO 200 J=1,NTH
      SL=S(IS,J)
      T(J)=TH(IS,J)
      L=MTH+1-J
      T(L)=PI-T(J)
```

```
      DO201 I=1,NS
      SPS(I)=SP(I,J)
      PH(I)=SPHI(I,J)
      FM(I)=SPHPP(I,J)
  201 DT(I)=DTP(I,J)
      CALL SPLEVN(SL,SPHIS(J),UP(IS,J),NS,TOL,SPS,PH,FM,DT)
      SPHIS(L)=SPHIS(J)
  200 CONTINUE

C   COMPUTE UT

      CALL PLINE(MTH,TOL,T,SPHIS,FM,DELTA)
      DO 202 J=1,MTH
      CALL SPLEVN(T(J),DUM,PHP(J),MTH,TOL,T,SPHIS,FM,DELTA)
  202 CONTINUE
      CALL PLINE (MTH,TOL,T,PHP,FM,DELTA)
      DO 203 J=1,MTH
      CALL SPLEVN(T(J),DUM,MY(J),MTH,TOL,T,PHP,FM,DELTA)
  203 CONTINUE
      DO 204 J=1,NTH
      CALL SPLEVN(T(J),DUM,UT(IS,J),MTH,TOL,T,PH,MY,DELTA)
  204 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SPOTEN(NS,NT,DTH,NTH)

C   SUBROUTINE USES SPLINE ON SPLINE INTERPOLATION TO FIT POTENTIAL
C         SP          S COORDINATES AT WHICH SPHI IS GIVEN
C         SPHI        SPLINE POTENTIAL
C         SPHP        DERIVATIVE OF SPHI IN S DIRECTION
C         SPHPP       SECOND DERIVATIVE OR  MOMENT OF SPHI
C         DTP         SPACING INCREMENTS IN S

      COMMON/SPLIN/SP(30,60),SPHI(30,60),SPHP(30,60),SPHPP(30,60),
     $ DTP(30,60)

C         T           ARCTAN(X/Y)
C         X           X COORDINATE
C         Y           Y COORDINATE
C         PHI         POTENTIAL FROM MING"S PROGRAM

      DIMENSION T(30,30),X(30,30),Y(30,30),PHI(30,30)

C         TL          SPLINE INTERPULATION COORDINATES
C         PH          SPECIFIED FUNCTION VALUES FOR SPLINE FIT
C         FM          MOMENTS
C         DT          SPACING
C         S           LONGITUDINAL POINTS AT WHICW POTENTIAL IS SPECIFIED
C         TH          THATA COORDINATES AT WHICH SPHI IS SPECIFIED
C         MY          MOMENTS

      DIMENSION TL(60),PH(60),FM(60),DT(60),S(60),MY(60)
      DIMENSION PHP(60),TH(60)
      REAL MY

C   READ INPUT AND CONVERT TO PROGRAMS COORDINATE SYSTEM

      READ (12,400) ((SP(I,J),X(I,J),Y(I,J),D,DU,DUM,PHI(I,J),J=1,NT)
     $,I=1,NS)
      WRITE(6,400) ((SP(I,J),X(I,J),Y(I,J),D,DU,DUM,PHI(I,J),J=1,NT)
     $,I=1,NS)
      PI=3.1415926
      TOL=1.0E-06
      LBC=3
      MT=2*NT
      TH(1)=0.0
      DO 200 I=1,NS
      DO 200 J=1,NT
      SP(I,J)=-SP(I,J)
```

```
      PHI(I,J)=-PHI(I,J)
  200 T(I,J)=ATAN2(X(I,J),Y(I,J))

C   BEGIN SPLINE ON SPLINE FIT OF POTENTIAL IN TRANSVERSE PLANES.

      DO 201 IS=1,NS
      DO 202 J=1,NT
      L=MT+1-J
      TL(J)=T(IS,J)
      PH(J)=PHI(IS,J)
      TL(L)=PI-TL(J)
  202 PH(L)=PH(J)
      CALL PLINE(MT,TOL,TL,PH,FM,DT)
      DO 203 J=1,MT
      CALL SPLEVN(TL(J),DUMMY,PHP(J),MT,TOL,TL,PH,FM,DT)
  203 CONTINUE
      CALL PLINE(MT,TOL,TL,PHP,FM,DT)
      DO 204 J=1,MT
      CALL SPLEVN(TL(J),BUMMY,MY(J),MT,TOL,TL,PHP,FM,DT)
  204 CONTINUE
      DO 201 J=1,NTH
      CALL SPLEVN(TH(J) ,SPHI(IS,J),DUMMY,MT,TOL,TL,PH,MY,DT)
  201 CONTINUE

C   BEGIN LONGITUDINAL SPLINE FIT

      DO 206 J=1,NTH
      DO 207 I=1,NS
      S(I)=SP(I,J)
      PH(I)=SPHI(I,J)
  207 CONTINUE
      CALL SPLINE(NS,TOL,S,PH,FM,DT)
      DO 208 I=1,NS
      CALL SPLEVN(S(I),DUMMY,PHP(I),NS,TOL,S,PH,FM,DT)
  208 CONTINUE
      CALL SPLINE(NS,TOL,S,PHP,FM,DT)
      DO 209 I=1,NS
      CALL SPLEVN(S(I),DUM,MY(I),NS,TOL,S,PHP,FM,DT)
  209 CONTINUE
      DO 206 I=1,NS
      CALL SPLEVN(S(I),SPHI(I,J),DUM,NS,TOL,S,PH,MY,DT)
      SPHP(I,J)=DUM
      SPHPP(I,J)=MY(I)
      DTP(I,J)=DT(I)
  206 CONTINUE
       RETURN
  400 FORMAT(8D10.4)
      END
```

```
      SUBROUTINE SPLINE(N,TOL,X,Y,M,H)
C     NONPERIODIC SPLINE FIT USING THE MOMENTS M(J) TO SPECIFY THE SPLIN
C     N NUMBER OF INTERPOLATION POINTS
C     TOL TOLERANCE ON SOLUTION
C     X MESH POINTS A=X(1).LT.X(2).LT. ... .LT.X(N)=B
C     Y ORDINATES Y(1)=F(X(1)),ETC.
C     H INTERVAL LENGTHS H(J)=X(J)-X(J-1)
C     LBC CONTROL NUMBER  TO PICK END CONDITIONS   LBC=
C     1 FIRST DERIVATIVE OF F AT X(1) AND AT X(N+1) IS GIVEN
C     2 M(1)=YOP,M(N)=YNP
C      3 M(1)=M(2),M(N-1)=M(N)
C     YOP DERIVATIVE AT X(1) OR VALUE OF M(1)
C     YNP DERIVATIVE AT X(N) OR VALUE OF M(N)
      COMMON/ENDCOD/LBC,YOP,YNP
      DIMENSION X(N),Y(N),M(N),H(N)
      DIMENSION L(99),MU(99),D(99)
      DIMENSION P(99),Q(99),U(99)
      DIMENSION DUMMYB(99),DC(99),TC(99)
      DIMENSION R(99),DT(99),UP(99),UN(99)
      REAL M,L,MU
      NMO=N-1
      DO 210 I=2,N
      H(I)=X(I)-X(I-1)
  210 CONTINUE
      DO 200 I=2,NMO
      L(I)=H(I+1)/(H(I)+H(I+1))
      MU(I)=1.0-L(I)
      D(I)=6.0*L(I)/H(I+1)*((Y(I+1)-Y(I))/H(I+1)-(Y(I)-Y(I-1))/H(I))
  200 CONTINUE
      H(N)=X(N)-X(N-1)
      IF(LBC-2) 300,301,302
  300 L(1)=1.0
      D(1)=6.0/H(2)*((Y(2)-Y(1))/H(2)-YOP)
      MU(N)=1.0
      D(N)=6.0/H(N)*(YNP-(Y(N)-Y(N-1))/H(N))
      GOTO 303
  301 L(1)=0.0
      D(1)=2.0*YOP
      MU(N)=0.0
      D(N)=2.0*YNP
      GOTO303
  302 L(1)=-2.0
      D(1)=0.0
      MU(N)=-2.0
      D(N)=0.0
```

```
  303 B=2.0
      P(1)=B
      Q(1)=-L(1)/P(1)
      U(1)=D(1)/P(1)
      L(N)=0.0
      R(1)=0.0
      DT(1)=D(1)/P(1)
      DO 201 I=2,N
      P(I)=MU(I)*Q(I-1)+B
      Q(I)=-L(I)/P(I)
      R(I)=MU(I)/P(I)
  201 DT(I)=D(I)/P(I)
      UP(1)=U(1)
      UN(1)=0.0
      DO 205 I=1,N
      UP(I)=DT(I)+R(I)*UN(I-1)
      UN(I)=R(I)*UP(I-1)
  205 U(I)=UP(I)-UN(I)
      M(N)=U(N)
      DO 202 I=2,N
      K=N+1-I
  202 M(K)=Q(K)*M(K+1)+U(K)
      DUMMYB(1)=B
      DC(1)= 2.0*M(1)+L(1)*M(2)
      TC(1)=D(1)-DC(1)
      DUMMYB(N)=B
      DC(N)=2.0*M(N)+MU(N)*M(N-1)
      TC(N)=D(N)-DC(N)
      DO 207 I=2,NMO
      DUMMYB(I)=B
      DC(I)=2.0*M(I)+MU(I)*M(I-1)+L(I)*M(I+1)
  207 TC(I)=D(I)-DC(I)
      K=1
      TL=D(1)-2.0*M(1)-L(1)*M(2)
      TL=ABS(TL)
      IF(TL.GE.TOL) GOTO 501
      K=N
      TL=D(N)-2.0*M(N)-MU(N)*M(N-1)
      TL=ABS(TL)
      IF(TL.GE.TOL) GOTO 501
      DO 203 I=2,NMO
      K=I
      T=D(I)-2.0*M(I)-MU(I)*M(I-1)-L(I)*M(I+1)
      T=ABS(T)
      TL=T
```

```
      IF(T.GE.TOL) GOTO 501
  203 CONTINUE
  500 RETURN
  501 WRITE(6,400)K,TL,TOL
      WRITE(9,400)K,TL,TOL
      MU(1)=-999999.9
      L (N)=-999999.9
      WRITE(6,401) (      MU(I),DUMMYB(I),L(I),M(I),D(I),DC(I),TC(I),
     1 I=1,N)
      RETURN
  400 FORMAT(I5,2E15.4)
  401 FORMAT(7E15.8)
      END
```

```
      SUBROUTINE PLINE(N,TOL,X,Y,M,H)
C    PERIODIC SPLINE FIT USING THE MOMENTS M(J) TO SPECIFY THE SPLINE
C     N NUMBER OF INTERPOLATION POINTS
C     TOL TOLERANCE ON SOLUTION
C     X MESH POINTS A=X(1).LT.X(2).LT.....LT.X(N)=B
C     Y ORDINATES Y(J)=F(X(J))
C     M MOMENTS M(J)=F""(X(J))
C      H INTERVAL LENGTHS H(J)=X(J)-X(J-1)
      DIMENSION X(N),Y(N),M(N),H(N)
      DIMENSION L(99),MU(99),D(99)
      DIMENSION P(99),Q(99),U(99),S(99),T(99),V(99)
      DIMENSION DC(99),TC(99)
      REAL M,L,MU
      TY=ABS (Y(1)-Y(N))
      IF(TY.GE.TOL) WRITE(6,403) Y(1),Y(N)
      NMO=N-1
      DO 210 I=2,N
      H(I)=X(I)-X(I-1)
  210 CONTINUE
      H(N+1)=H(2)
      Y(N+1)=Y(2)
      DO 200 I=2,N
      L(I)=H(I+1)/(H(I)+H(I+1))
      MU(I)=1.0-L(I)
      D(I)=6.0*L(I)/H(I+1)*((Y(I+1)-Y(I))/H(I+1)-(Y(I)-Y(I-1))/H(I))
  200 CONTINUE
      B=2.0
      P(2)=B
      Q(2)=-L(2)/P(2)
      U(2)=D(2)/P(2)
      S(2)=-MU(2)/P(2)
      DO 201 I=3,N
      P(I)=MU(I)*Q(I-1)+B
      Q(I)=-L(I)/P(I)
      S(I)=-MU(I)*S(I-1)/P(I)
  201 U(I)=(D(I)-MU(I)*U(I-1))/P(I)
      T(N)=1.0
      V(N)=0.0
      DO 205 I=2,NMO
      K=N+1-I
      T(K)=Q(K)*T(K+1)+S(K)
      V(K)=Q(K)*V(K+1)+U(K)
  205 CONTINUE
      C=L(N)*T(2)+MU(N)*T(N-1)+B
      DV=L(N)*V(2)+MU(N)*V(N-1)
```

```
      M(N)=(D(N)-DV)/C
      DO 202 I=2,NMO
      K=N+1-I
  202 M(K)=T(K)*M(N)+V(K)
      M(1)=M(N)
      M(N+1)=M(2)
      DO 206 I=2,N
      DC(I)=2.0*M(I)+MU(I)*M(I-1)+L(I)*M(I+1)
  206 TC(I)=D(I)-DC(I)
      DO 203 I=2,N
      K=I
      TL=TC(I)
      TL=ABS (TL)
      IF(TL.GE.TOL) GOTO 501
  203 CONTINUE
      RETURN
  501 WRITE(6,400)K,TL,TOL
      WRITE(6,401) (M(I),D(I),DC(I),TC(I),I=2,N)
      RETURN
  400 FORMAT(I5,2E15.4)
  401 FORMAT(4E15.8)
  403 FORMAT(40H2Y(1) IS NOT EQUAL Y(N), NOT PERIODIC        ,2E15.8)
      END
```

```
      SUBROUTINE SPLEVN(XE,YE,YP,N,TOL,X,Y,M,H)
C  EVALUATES A SPLINE INTERPOLATION AT XE WHEN THE MOMENTS M(J) ARE
C     GIVEN AT X(J).
C     YE      IS THE VALUE OF THE SPLINE AT XE
C     H(J)    INTERVAL LENGTHS
C     N       NUMBER OF MESH POINTS
C     Y(J)    INTERPOLATION VALUES AT INTERPOLATION POINTS X(J) RESP.
      DIMENSION X(N),Y(N),M(N),H(N)
      REAL M
      J=1
  300 J=J+1
      IF(J  .GT.N) GOTO 501
      IF(XE.GT.X(J))GO TO 300
      YE=(M(J-1)*((X(J)-XE)**3)+M(J)*((XE-X(J-1))**3))/6.0
      YE=YE+(Y(J-1)-M(J-1)*H(J)*H(J)/6.0)*(X(J)-XE)+(Y(J)-M(J)*H(J)*
     1 H(J)/6.0)*(XE-X(J-1))
      YE=YE/H(J)
      YP=(-M(J-1)*(X(J)-XE)*(X(J)-XE)+M(J)*(XE-X(J-1))*(XE-X(J-1)))*0.5
      YP=(YP+Y(J)-Y(J-1))/H(J)
      YP=YP-(M(J)-M (J-1))*H(J)/6.0
      RETURN
  501 WRITE(6,400) XE,X(N)
      RETURN
  400 FORMAT(4H XE,1E10.4,13H ,EXCEEDS DX     ,1E10.4)
      END
```

```
#FILE (CHXL)TBLSOL ON DTNSRDC
 FILE  5(KIND=REMOTE,MAXRECSIZE=22)
 FILE  6(KIND=PRINTER,MAXRECSIZE=132)
 FILE  7(TITLE="TBLSOLOPT",KIND=DISK,PROTECTION=SAVE,MAXRECSIZE=22)
 FILE  8(TITLE="TBLIPCOPT",KIND=DISK,FILETYPE=7)
 FILE  9(KIND=REMOTE,MAXRECSIZE=22)
 FILE  10(TITLE="WING3DOPT",KIND=DISK,FILETYPE=7)
 FILE  11(TITLE="THHBETA",KIND=DISK,FILETYPE=7)
  $RESET FREE
 C      PROGRAM TBLSOL
 C      PROGRAM TO SOLVE PARTIAL DIFFERENTIAL EQUATIONS . USES THE O BRIEN
 C     $ HYMAN,AND KAPLAN IMPLICIT FORMULATION
 C      THE RESULTING SYSTEM OF LINEAR EQUATIONS ARE SOLVED USING A
 C     $ DIRECT GAUSSIAN REDUCTION APPLIED TO THE SPARSE COEFFICIENT MATRI
        COMMON/INT/NTH,MTH,NS,DELTAT
        COMMON/VAR/XV
        COMMON/CDOD/ODL,D,ODR,CR
        COMMON/COEFPC/PC
        COMMON/DATAIN/ TH,UP,UT,S,UPLP,UTLP,UPLT,UTLT
        COMMON/GEOIN/KP,KT,H,F,GG
        COMMON/JON/JO,JN
        COMMON/TOLERN/TOL
        REAL KP,KT
        DIMENSION TITLE(12),TITL(3)
        DIMENSION X(2,150),Y(2,150),        UP(2,150),UT(2,150)
        DIMENSION TH(2,150)
        DIMENSION PC(2,4)
        DIMENSION XV(150,3)
        DIMENSION S(2,150),UPLP(2,150),UTLP(2,150),UPLT(2,150),UTLT(2,150)
        DIMENSION KP  (2,150),KT (2,150),H(2,150),F(2,150),GG(2,150)
        DIMENSION D(150,3,3),ODL(150,3,3),ODR(150,3,3),CR(150,3)
        DATA((PC(I,J),J=1,4),I=1,2)/
       $-0.000701,0.028345,-0.386768,0.019521,
       $ -0.001953,0.062588,-0.83480,0.191511/
        WRITE(9,411)((PC(I,J),J=1,4),I=1,2)
        WRITE(9,412)
        WRITE(9,413)
        WRITE(9,414)
        READ(5,415) NS
        WRITE(9,415) NS
        WRITE(9,416)
        WRITE(9,417)
        READ (5,415) NI
        WRITE(9,415) NI
        WRITE(6,410) NS,NI
        DATA TOL/0.100E-09/
```

```
      WRITE(9,409) TOL
C     INPUT PRELIMINARY DATA FROM TAPE
      READ (8,401) (TITLE(J),J=1,12)
      WRITE(6,401) (TITLE(J),J=1,12)
      WRITE(7,401) (TITLE(J),J=1,12)
      READ (8,402 ) N,KPK,BL
      WRITE(6,402 ) N,KPK,BL
      WRITE(7,402 ) N,KPK,BL
      READ (8,405) START,SFINS,NSTAT,NTH
      WRITE(6,405) START,SFINS,NSTAT,NTH
      WRITE(7,405) START,SFINS,NSTAT,NTH
      MTH=NTH-1
C     SUBROUTINE INITIAL DEFINES THE INITIAL DATA
      CALL INITAL
      JO=1
      JN=2
      READ (8,427)JJ,SS
      WRITE(6,427)JJ,SS
      READ (8,427) JJ,SS
      WRITE(6,427) JJ,SS
      WRITE(7,427) JJ,SS
C     ENTER GEOMETRY AND POTENTIAL DATA FOR INITIAL LINE.
      DO 200 ITH=1,NTH
      READ (8,408) NDUM,X(1,ITH),Y(1,ITH),TH(1,ITH),UP(1,ITH),UT(1,ITH)
      WRITE(6,408) NDUM,X(1,ITH),Y(1,ITH),TH(1,ITH),UP(1,ITH),UT(1,ITH)
      WRITE(7,408) NDUM,X(1,ITH),Y(1,ITH),TH(1,ITH),UP(1,ITH),UT(1,ITH)
      READ (8,408) NDUM,S(1,ITH),UPLP(1,ITH),UPLT(1,ITH),UTLP(1,ITH),
     $ UTLT(1,ITH)
      WRITE(6,408) NDUM,S(1,ITH),UPLP(1,ITH),UPLT(1,ITH),UTLP(1,ITH),
     $ UTLT(1,ITH)
      WRITE(7,408) NDUM,S(1,ITH),UPLP(1,ITH),UPLT(1,ITH),UTLP(1,ITH),
     $ UTLT(1,ITH)
      READ (8,408) NDUM,KP(1,ITH),KT(1,ITH),H(1,ITH),F(1,ITH),GG(1,ITH)
      WRITE(6,408) NDUM,KP(1,ITH),KT(1,ITH),H(1,ITH),F(1,ITH),GG(1,ITH)
      WRITE(7,408) NDUM,KP(1,ITH),KT(1,ITH),H(1,ITH),F(1,ITH),GG(1,ITH)
  200 CONTINUE
      DELTAT=TH(1,2)
      DO 201 IS=1,NSTAT
       MS=IS
      IF(IS.GT.NS) GO TO 500
      READ(8,427) JJ,SS
      WRITE(6,427) JJ,SS
      DO 202 IT=1,NTH
C     ENTER GEOMETRY AND POTENTIAL DATA FOR PHI-STATION JN.
      READ (8,408) NDUM,X(JN,IT),Y(JN,IT),TH(JN,IT),UP(JN,IT),UT(JN,IT)
```

```
      READ (8,408) NDUM,S(JN,IT),UPLP(JN,IT),UPLT(JN,IT),UTLP(JN,IT)
     $ ,UTLT(JN,IT)
      READ (8,408) NDUM,KP(JN,IT),KT(JN,IT),H(JN,IT),F(JN,IT),GG(JN,IT)
  202 CONTINUE
      DO 205 II=1,NI
C     COMPUTE COEFFICIENTS FOR LINEAR EQUATIONS
C     SUBROUTINE  ZSYPLN COMPUTES THE COEFFICIENT ALONG THE SYMMETRY
C    $ LINE THATA EQUAL ZERO, THE LOAD WATER LINE.
      CALL ZSYPLN
C
C
C     SUBROUTINE LECOEF COMPUTES THE COEFFICIENTS OF THE LINEAR
C     EQUATIONS EXCEPT ON THE SYMMETRY LINES.
C
C
      CALL LECOEF
  203 CONTINUE
C
C     SUBROUTINE FSYPLN COMPUTES THE COEFFICIENTS ON THE SYMMETRY PLANE
C     THATA = PI/2.
C
      CALL FSYPLN
C     SOLUTION OF LINEAR EQUATIONS IS XV
C     SUBROUTINE SOLVE COMPUTES THE SOLUTION.
      CALL SOLVE
  602 CONTINUE
      WRITE(6,404)
      K=JN
      DO 204 ITH=1,NTH
      WRITE(6,403)TH(K,ITH),X(K,ITH),Y(K,ITH),S(K,ITH),(XV(ITH,I),I=1,3)
      WRITE(7,403)TH(K,ITH),X(K,ITH),Y(K,ITH),S(K,ITH),(XV(ITH,I),I=1,3)
  204 CONTINUE
  205 CONTINUE
      JO=JN
      IF(JO.EQ.1) JN=2
      IF(JO.EQ.2) JN=1
  201 CONTINUE
  500 CONTINUE
  401  FORMAT(12A6)
  402 FORMAT(2I10,1F10.5)
  403 FORMAT( 4F10.4,3E20.8)
  404  FORMAT(1H2,4X,6H THATA,6X,2H X,9X,2H Y,9X,2H S,11X,5H PHI,15X,
     $  2H T,19X,2H H)
  405  FORMAT(2F10.5,2I10)
  406 FORMAT(10X,3E15.7)
```

```
  407 FORMAT(5X,3E15.7)
  408  FORMAT(1I3,5E15.7)
  409 FORMAT(1E14.5)
  410 FORMAT(5I4       )
  411  FORMAT(4F10.6)
  412 FORMAT(54H NS IS NUMBER OF STATIONS AT WHICH BOUNDARY LAYER IS   )
  413 FORMAT(19H TO BE CALCULATED    )
  414 FORMAT(19H NS=   1I3           )
  415 FORMAT(1I3)
  416 FORMAT(43H NI IS NUMBER OF ITTERATIONS AT EACH STEP      )
  417 FORMAT(16H NI=     1I3        )
  421  FORMAT(26H INPUT DATA OUT OF ORDER.   /2I8)
  427  FORMAT(1I3,1F9.6)
  470  FORMAT(3A10)
       END
```

```
      SUBROUTINE INVERS(A,AI,T)
C     COMPUTES INVERSE OF 3*3 MATRIX A
C     AI IS INVERSE  TIS TOLERANCE
      DIMENSION A(3,3),AI(3,3)  ,TM(3,3),EM(3,3)
      DET=A(1,1)*A(2,2)*A(3,3)+A(1,2)*A(2,3)*A(3,1)+A(2,1)*A(3,2)*A(1,3)
      D=A(1,1)*A(2,3)*A(3,2)+A(1,2)*A(2,1)*A(3,3)+A(1,3)*A(2,2)*A(3,1)
      DET=DET-D
      DAB=ABS  (DET)
      IF(DAB.LT.T) GO TO 501
      DI=1.0/DET
      AI(1,1)= A(2,2)*A(3,3)-A(3,2)*A(2,3)
      AI(2,1)= A(2,3)*A(3,1)-A(2,1)*A(3,3)
      AI(3,1)=-A(2,2)*A(3,1)+A(2,1)*A(3,2)
      AI(1,2)=+A(1,3)*A(3,2)-A(1,2)*A(3,3)
      AI(3,2)=-A(1,1)*A(3,2)+A(1,2)*A(3,1)
      AI(2,2)= A(1,1)*A(3,3)-A(1,3)*A(3,1)
      AI(1,3)= A(1,2)*A(2,3)-A(1,3)*A(2,2)
      AI(2,3)=-A(1,1)*A(2,3)+A(1,3)*A(2,1)
      AI(3,3)= A(1,1)*A(2,2)-A(1,2)*A(2,1)
      DO 200 I=1,3
      DO 200 J=1,3
      AI(I,J)=DI*AI(I,J)
  200 CONTINUE
      DO 201 I=1,3
      DO 201 J=1,3
      TM(I,J)=0.0
      EM(I,J)=0.0
      IF(I.EQ.J) EM(I,J)=1.0
      DO 201 K=1,3
  201 TM(I,J)=TM(I,J)+A(I,K)*AI(K,J)
      DO 202 I=1,3
      DO 202 J=1,3
      IS=I
      JS=J
      DTM=TM(I,J)-EM(I,J)
      ATM=ABS(DTM)
      IF(ATM.GT.T) GO TO 502
  202 CONTINUE
  500 RETURN
  501 WRITE(6,400) DET,T
      WRITE(6,401) A
      STOP
  502 WRITE(6,400) DET,T
      WRITE(6,401) A
      WRITE(6,401) AI
```

```
      WRITE(6,403) IS,JS
      WRITE(6,401) TM
      WRITE(6,401) EM
      WRITE(6,402)
      STOP
  400 FORMAT(5H2DET=,1E15.7,5X,3H T=,1E15.7 //2H A//)
  401 FORMAT(3E15.7)
  402 FORMAT(25H INVERS IS NOT CORRECT        )
  403 FORMAT(2I5)
      END
```

```
      SUBROUTINE INITAL
C     DEFINES INITIAL CONDITIONS
      COMMON/VAR/XV
      COMMON/INT/NTH,MTH.MS,DELTAT
      DIMENSION XV(150,3)
      WRITE(9,402)
      WRITE(9,403) <— WRITE (9,411)
      READ(5,404) NC
      IF(NC-1)100,101,102
  100 WRITE(9,405)
      READ (5,401) TH11
      WRITE(9,401) TH11
      WRITE(9,406)
      READ (5,401) T
      WRITE(9,401) T
      WRITE(9,407)
      READ (5,401) H
      WRITE(9,401) H
      WRITE(6,401)TH11,T,H
      DO 200 ITH=1,NTH
      XV(ITH,1)=TH11
      XV(ITH,2)=T
  200 XV(ITH,3)=H
      WRITE(6,400) TH11,T,H
      WRITE(7,401) TH11,T,H
      RETURN
  101 READ (11,408) NTHCK
      IF(NTH.NE.NTHCK) GOTO 300
      READ (11,401) ((XV(I,J),J=1,3),I=1,NTH)
      WRITE( 6,401) ((XV(I,J),J=1,3),I=1,NTH)
      WRITE( 7,401) ((XV(I,J),J=1,3),I=1,NTH)
      RETURN
  300 WRITE(9,408)NTH,NTHCK
      WRITE(9,409)
      STOP
  102 WRITE(9,410)
      STOP
  400 FORMAT(16H1INITIAL VALUES      //6H TH11=1F10.4,5X,3H T= 1F10.4,
     $  5X,3H H=1F10.4///)
  401 FORMAT(3F10.5)
  402 FORMAT(45H IF THE INITIAL VALUES ARE CONSTANT ENTER 0    )
  403 FORMAT(22H OTHERWISE ENTER A X    ) 2
  404 FORMAT(1I1)
  405 FORMAT(19H TH11=      1F10.5    )
  406 FORMAT(30H T= TAN(BETA)=      1F10.5     )
```

411 Format (if the Initial Values
are specified as a function of them Enter 1)

```
  407 FORMAT(19H H=         1F10.5    )
  408 FORMAT(5H NTH=,1I3,/7H NTHCK=,1I3 ///)
  409 FORMAT(48H NUMBER OF INITIAL DATA NOT COMPATIBLE WITH NTH  )
  410 FORMAT(47H WRITE OWN ROUTINE FOR ENTERING INITIAL DATA        )
      END
```

```
      SUBROUTINE COEFF(TH,H,SA,CA,T)
C     ALL FUNCTIONS COMPUTED IN THIS SUBROUTINE ARE FUNCTIONS OF THATA.
C     H, ALPHA, AND T
C     SA=SIN(ALPHA), CA=COS (ALPHA)
C     A IS THE COEFFICIENT MATRIX FROM THE LEFT SIDE OF THE PDE
C     B IS THE COEFFICIENT MATRIX FROM THE RIGHT SIDE
C     C IS THE FORCING FUNCTION
C     VARIABLES BEGINNING WITH G OR H ARE DEFINED IN SUBROUTINE GS
C    PC ARE THE COEFFICIENTS OF THE PILYNOMIALS A AND B (NOT RELATED
C     TO COEFFICIENT MATRICES) WHICH BRING IN THE REYNOLDS NUMBER
C     DEPENDENCE.
      COMMON/CH/FG,DGDH
      COMMON/CABC/A,B,C
      COMMON/CG/G,H1,H2,TFG,SFG
      COMMON/CGD/GH,GA,GT,H1T,H1H,H2T,H2H
      COMMON/CGDD/GHH,GHA,GHT,GAA,GAT,GTT
      COMMON/CINPUT/ULP,VLP,ULTH,VLTH,UM,KP,KT,UMLTH,UMLP
      COMMON/COEFPC/PC
      COMMON MT
      REAL KP,KT
      DIMENSION A(3,3),B(3,3),C(3)
      DIMENSION G(2,2),GH(2,2),GA(2,2),GT(2,2)
      DIMENSION GHH(2,2),GHT(2,2),GHA(2,2),GAA(2,2),GAT(2,2),GTT(2,2)
      DIMENSION PC(2,4)
      UMLT=UMLTH
      CALL GS(H,SA,CA,T,1)
C     COMPUTE A
  600 A(1,1)=G(1,1)
      A(1,2)=TH* GT(1,1)
      A(1,3)=TH*GH(1,1)
      A(2,1)=G(2,1)
      A(2,2)=TH*GT(2,1)
      A(2,3)=TH*GH(2,1)
      A(3,1)=TFG
      A(3,2)=TH*H1T
      A(3,3)=TH*H1H
C     COMPUTE B
      B(1,1)=G(1,2)
      B(1,2)=TH*GT(1,2)
      B(1,3)=TH*GH(1,2)
      B(2,1)=G(2,2)
      B(2,2)=TH*GT(2,2)
      B(2,3)=TH*GH(2,2)
      B(3,1)=SFG
      B(3,2)=TH*H2T
```

```
      B(3,3)=TH*H2H
      IF(MT.NE.3) GOTO 100
      WRITE(6,400) FG,DGDH,TH,H,T,SA,CA
      WRITE(6,403) G
      WRITE(6,403) GH
       WRITE(6,403)GT
      WRITE(6,402) A
      WRITE(6,402) B
  100 CONTINUE
C     REYNOLDS NUMBER INFLUENCE
C     REYNOLD"S NUMBER BASED ON LENGTH IS SET AT 10**6
      RL=1.0E 06
      RT=RL*TH*UM
      RT=ABS(RT)
      RC=ALOG(RT)
      PA=PC(1,1)
      PB=PC(2,1)
      DO 200 J=2,4
      PA=RC*PA+PC(1,J)
  200 PB=RC*PB+PC(2,J)
      CF1=EXP (PA*H+PB)
      CFP=CF1*(CA+T*SA)
      CFT=CF1*(-SA+T*CA)
      DELTA1=TH*H1
      DELTA2=TH*H2
      TF=TH*TFG
      SF=TH*SFG
C     COMPUTE C
      UMI=1.0/(1.0+UM)
      C(1)= 0.5*CFP-2.0*TH*UMI*(G(1,1)*UMLP+G(1,2)*UMLT)
      IF(MT.EQ.3) WRITE(6,401) C(1)
      C(1)= C(1)-UMI*(DELTA1*ULP+DELTA2*ULTH)
      IF(MT.EQ.3) WRITE(6,401) C(1)
      XKT=G(1,2)+G(2,1)+SA*H1
      XKP=G(1,1)-G(2,2)-SA*H2
      IF(MT.NE.3)GO TO 300
      WRITE(6,400) PA,PB,CF1,CFP,CFT
      WRITE(6,400) DELTA1,DELTA2,TF,SF,UMI
      WRITE(6,400) XKT,XKP
  300 CONTINUE
      C(1)= C(1)+TH*(KP*XKP+KT*XKT)
      IF(MT.EQ.3) WRITE(6,401) C(1)
      C(2)=0.5*CFT-2.0*UMI*(G(2,1)*UMLP+G(2,2)*UMLT)*TH
      C(2)= C(2)-UMI*(DELTA1*VLP+DELTA2*VLTH)
      XKP=G(1,2)+G(2,1)+CA*H2
```

```
      XKT=G(2,2)-G(1,1)-CA*H1
      C(2)= C(2)+TH*(KP*XKP+KT*XKT)
      C(3)=0.025*H-0.022
      IF(MT.EQ.3) WRITE(6,401) C(3)
      UMI=1.0/UM
      C(3)= C(3)-UMI*(TF*UMLP+SF*UMLT)
      IF(MT.EQ.3) WRITE(6,401) C(3)
      C(3)= C(3)+TF*KP+SF*KT
      IF(MT.EQ.3) WRITE(6,401) C(3)
  500 RETURN
  400 FORMAT(3X,5E15.6)
  401 FORMAT(3H C= ,1E15.6)
  402 FORMAT(3E15.7)
  403 FORMAT(2E15.7)
CATALOG(BLTDMY,BTBLCOEF,ID=CHXL,RP=120)
      END
```

```
      SUBROUTINE GS(H,SA,CA,T,MG)
C     ALL FUNCTIONS COMPUTED IN THIS SUBROUTINE ARE FUNCTIONS OF H,
C    $ ALPHA AND T
C     SA IS SIN (ALPHA) AND CA IS COS(ALPHA)
C     G(I,J) ARE THE FUNCTIONS G SUBSCRIP IJ
C     GH IS THE DERIVATIVE OF G WITH RESPECT TO H
C     GA ""     ""     ""     ""     ""     ""  ALPHA
C     GT ""     ""     ""     ""     ""     ""     T
C     GHH,GHT,GHA,GTT,GTA,GAA ARE THE RESPECTIVE SECOND PARTIAL
C    $ DERIVATIVES.
      COMMON/CF/FB
      COMMON/CFD/FBP,FBPP
      COMMON/CG/G,H1,H2,TFG,SFG
      COMMON/CGD/GH,GA,GT,H1T,H1H,H2T,H2H
      COMMON/CFG/FG,DGDH
      DIMENSION FB(4),FBP(4),FBPP(4)
      DIMENSION G(2,2),GH(2,2),GA(2,2),GT(2,2)
      CALL FS(H)
C     COMPUTE H1,H2,H11, AND H22
C     H11 WILL BE COMPUTED AS TFG
C     H22 WILL BE COMPUTED AS SFG
C     H1T,H1H,H2T,H2H ARE PARTIAL DERIVATIVES WITHWITH RESPECT TO T AND
C   $ H OF THE FUNCTIONS H11 AND H22
      H1T=-FB(4)*SA
      H2T=FB(4)*CA
      H1=H*CA+T*H1T
      H2=H*SA+T*H2T
C     COMPUTE THE FUNCTION T=G*COS(ALPHA)+T*FB4*SIN(ALPHA)
C     COMPUTE THE FUNCTION         S=G*SIN(ALPHA)-T*FB4*COS(ALPHA)
      FG=2.0*H/(H-1.0)
      TFG=FG*CA+T*FB(4)*SA
      SFG=FG*SA-T*FB(4)*CA
      H1T=-H1T
      H2T=-H2T
      DGDH=-2.0/((H-1.0)*(H-1.0))
      H1H=CA*DGDH+T*FBP(4)*SA
      H2H=SA*DGDH-T*FBP(4)*CA
C     COMPUTATION OF G"S
      CAS=CA*CA
      SAS=SA*SA
      SACA=SA*CA
      TS=T*T
      FOFT=FB(1)+FB(2)
      G(1,1)=CAS-T*FOFT*SACA+TS*FB(3)*SAS
      P=1.0-TS*FB(3)
```

```
      G(1,2)=P*SACA+T*(FB(2)*CAS-FB(1)*SAS)
      G(2,1)=P*SACA+T*(FB(1)*CAS-FB(2)*SAS)
      G(2,2)=SAS+T*FOPT*SACA    +TS*FB(3)*CAS
C     COMPUTE FIRST DERIVATIVES
      CSMSS=CAS-SAS
      GT(1,1)= FOPT*SACA+2.0*T*FB(3)*SAS
      GH(1,1)=-T*(FBH(1)+FBH(2)) [illegible] SACA+TS [illegible] *SA [illegible]
      GT(1,2) [illegible] 2.0*T*FB(3)*SACA [illegible] B(2)*CAS [illegible]
      GH(1,2)= [illegible] TS*FBH(3)*SACA+T* [illegible] FBH(2) [illegible] AS [illegible]
      GA(2,1)=P*CSM [illegible] *SACA
      GT(2, [illegible] )=-2.0*T*FB(3)*SACA [illegible]
      GH(2,1)= [illegible] TS*FBH(3)*SACA [illegible]
      GT(2,2)=FOPT*SACA [illegible]
      GH(2,2) [illegible] SACA [illegible]
  500 RETURN
      END
```

```
      SUBROUTINE [illegible]COEF
C     COMPUTES THE COEFFICIENTS OF THE LINEAR EQUATIONS RESULTING FROM
C      THE IMPLICIT FORMULATION OF THE PDE.
      COMMON INT [illegible],DELTAT
      COMMON [illegible],DDR,[illegible]
      COMMON VAR[illegible]
      COMMON COEF[illegible]
      COMMON DATAIN [illegible]
      COMMON [illegible]
      COMMON [illegible]
      COMMON [illegible]
      COMMON [illegible]
      COMMON [illegible]
      COMMON AT
      [illegible]
      DIMENSION [illegible]
      DIMENSION [illegible],4[illegible]
      DIMENSION [illegible]
      DIMENSION [illegible]
      DIMENSION [illegible],150[illegible],150[illegible]
      DIMENSION [illegible],3,3[illegible]150[illegible]
      DIMENSION [illegible]
      DIMENSION [illegible],AS[illegible]3,3[illegible]
      [illegible]
      AT[illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      WRITE[illegible]
      WRITE[illegible]
```

```
      WRITE(6,400) XKP,XKT
      WRITE(6,406) A
      WRITE(6,407) B
      WRITE(6,406) C
  301 CONTINUE
      ADEL=ABS(        DELTAT)
      IF(ADEL.LE.TOL) GO TO 500
      IF(H(JN,IT).LT.TOL) GO TO 300
      IF(GG(JN,IT).LT.TOL) GO TO 300
      SS=S(JN,IT)-S(JO,IT)
      SST=0.5*SS/DELTAT
      HI=1.0/H(JN,IT)
      GGI=1.0/GG(JN,IT)
      DO 201 IS=1,3
      C(IS)=SS*C(IS)
      DO 201 JS=1,3
      B(IS,JS)=SST*B(IS,JS)*GGI
  201 AS(IS,JS)=HI*A(IS,JS)
      CALL INVERS(AS,ASI,TOL)
      IF(IT.NE.3) GO TO 302
      WRITE(6,407) AS
      WRITE(6,406) ASI
      WRITE(6,407) B
      WRITE(6,406) C
  302 CONTINUE
      DO 202 IB=1,3
      DO 202 JB=1,3
      ODR(IT,IB,JB)=0.0
      DO 202 K=1,3
  202 ODR(IT,IB,JB)=ODR(IT,IB,JB)+ASI(IB,K)*B(K,JB)
      DO 203 IB=1,3
      CR(IT,IB)=XV(IT,IB)
      DO 203 K=1,3
  203 CR(IT,IB)=CR(IT,IB)+ASI(IB,K)*C(K)
  200 CONTINUE
  500 RETURN
  300 WRITE(6,401)
      WRITE(6,402) JN,MT,DELTAT,H(JN,MT),GG(JN,MT)
  400 FORMAT(10X,5E15.7)
  401 FORMAT(5X, 30H DELTAT,H, OR GG ARE ZERO.                )
  402 FORMAT(5X,2I5,3E15.7)
  403 FORMAT(16H3FROM LECOEF MT=  ,I4//10X,5H TH11,15X,2H T,15X,2H H)
  406 FORMAT(15X,3E15.7    )
  407 FORMAT (5X,3E15.6)
      END
```

```
      SUBROUTINE ZSYPLN
C     COMPUTES THE COEFFICIENTS OF THE LINEAR EQUATIONS RESULTING FOOM
C      THE IMPLICIT FORMULATION OF THE PDE ON THE LINE OF SYMMETRY
C     THATA= 0.0
      COMMON/INT/NTH,MTH,NS,DELTAT
      COMMON/CDOD/ODL,D,ODR,CR
      COMMON/VAR/XV
         COMMON/COEFPC/PC,RL
      COMMON/DATAIN/ TH,UP,UT,S,UPLP,UTLP,UPLT,UTLT
      COMMON/GEOIN/KP.KT,H,F,GG
      COMMON/CABC/       A,B,C
      COMMON/CINPUT/ULP,VLP,ULT ,VLT,UM,XKP,XKT,UMLT,UMLP
      COMMON/JON/JO,JN
      REAL KP,KT
      DIMENSION XV(150,3)
      DIMENSION PC(2,4)
      DIMENSION UP(2,150),UT(2,150),TH(2.150)
      DIMENSION S(2,150),UPLP(2,150),UTLP(2,150),UPLT(2,150),UTLT(2,150)
      DIMENSION KP  (2,150),KT (2,150),H(2,150),F(2,150),GG(2,150)
      DIMENSION D(150,3,3),ODL(150,3,3),ODR(150,3,3),CR(150,3)
      DIMENSION A(3,3),B(3,3),C(3)
      DIMENSION AS(3,3). ASI(3,3)
      IT=1
      MT=IT
      ALPHA=ATAN2(UT(JN,IT),UP(JN,IT))
      UM=UT(JN,IT)*UT(JN,IT)+UP(JN,IT)*UP(JN,IT)
      UM=SQRT(UM)
      SA=UT(JN,IT)/UM
      CA=UP(JN,IT)/UM
      ULP=UPLP(JN,IT)
      VLP=UTLP(JN,IT)
      ULT=UPLT(JN,IT)
      VLT=UTLT(JN,IT)
      XKP=KP(JN,IT)
      XKT=KT(JN,IT)
      UMLT=CA*ULT +SA*VLT
      UMLP=CA*ULP+SA*VLP
      CALL COEFF(XV(IT,1),XV(IT,3),SA,CA,XV(IT,2))
      IF(H(JN,IT).LT.0.000001) GO TO 300
      IF(GG( JN,IT).LT.0.000001) GO TO 300
      SS=S(JN,IT)-S(JO,IT)
      SST=0.5*SS/DELTAT
      SST=2.0*SST
      HI=1.0/H(JN,IT)
      GGI=1.0/GG(JN,IT)
```

```
      DO 201 IS=1,3
      CR(IT IS)=0.0
      C(IS)=SS*C(IS)
      DO 201 JS=1,3
      ODR(IT,IS,JS)=0.0
      B(IS,JS)=SST*B(IS,JS)*GGI
  201 AS(IS,JS)=HI*A(IS,JS)
      DET=AS(1,1)*AS(3,3)-AS(1,3)*AS(3,1)
      DETI=1.0/DET
      ODR(IT,1,2)= DETI*(AS(3,3)*B(1,2)-AS(1,3)*B(3,2))
      ODR(IT,3,2)=-DETI*(AS(3,1)*B(1,2)-AS(1,1)*B(3,2))
      CR(IT,1)=XV(IT,1)+DETI*(AS(3,3)*C(1)-AS(1,3)*C(3))
      CR(IT,2)=XV(1,2)
      CR(IT,3)=XV(IT,3)-DETI*(AS(3,1)*C(1)-AS(1,1)*C(3))
  500 RETURN
  300 WRITE(6,401)
      WRITE(6,402) JN,MT,DELTAT,H( JN,MT),GG(JN,MT)
      STOP
  400 FORMAT(10X,5E15.7)
  401 FORMAT(5X, 30H DELTAT,H, OR GG ARE ZERO.
  402 FORMAT(5X,2I5,3E15.7)
  407 FORMAT (5X,3E15.6)
      END
```

```
      SUBROUTINE FSYPLN
C     COMPUTES THE COEFFICIENTS OF THE LINEAR EQUATIONS RESULTING FROM
C      THE IMPLICIT FORMULATION OF THE PDE ON THE LINE OF SYMMETRY
C     THATA= 0.0
      COMMON/INT/NTH,MTH,NS,DELTAT
      COMMON/CDOD/ODL,D,ODR,CR
      COMMON/VAR/XV
         COMMON/COEFPC/PC,RL
      COMMON/DATAIN/ TH,UP,UT,S,UPLP,UTLP,UPLT,UTLT
      COMMON/GEOIN/KP,KT,H,F,GG
      COMMON/CABC/      A,B,C
      COMMON/CINPUT/ULP,VLP,ULT ,VLT,UM,XKP,XKT,UMLT,UMLP
      COMMON/JON/JO,JN
      REAL KP,KT
      DIMENSION XV(150,3)
      DIMENSION PC(2,4)
      DIMENSION UP(2,150),UT(2,150),TH(2,150)
      DIMENSION S(2,150),UPLP(2,150),UTLP(2,150),UPLT(2,150),UTLT(2,150)
      DIMENSION KP  (2,150),KT (2,150),H(2,150),F(2,150),GG(2,150)
      DIMENSION D(150,3,3),ODL(150,3,3),ODR(150,3,3),CR(150,3)
      DIMENSION A(3,3),B(3,3),C(3)
      DIMENSION AS(3,3), AS[illegible](3,3)
      IT=NTH
      MT=IT
      ALPHA=A[illegible](JN,IT),UP(JN,IT))
      UM=UT(JN,IT)*UT(JN,IT)+UP(JN,IT)*UP(JN,IT)
      UM=SQRT(UM)
      SA=UT(JN,IT)/UM
      CA=UP(JN,IT)/UM
      ULP=UPLP(JN,IT)
      VLP=UTLP(JN,IT)
      ULT=UPLT(JN,IT)
      VLT=UTLT(JN,IT)
      XKP=KP(JN,IT)
      XKT=KT(JN,IT)
      UMLT=CA*ULT [illegible]SA*VLT
      UMLP=CA*UL[illegible]
      CALL COEFF(X[illegible]    XV(IT,3),SA,CA,XV(IT,2))
      IF(H(JN,IT).LT.0.000001) GO TO 300
      IF(GG( JN,IT).LT.0.000001) GO TO 300
      SS=S(JN,IT)-S(JO,IT)
      SST=0.5*SS/DELTAT
      SST=2.0*SST
      HI=1.0/H(JN,IT)
      GGI=1.0/GG(JN,IT)
```

```
      DO 201 IS=1,3
      CR(IT,IS)=0.0
      C(IS)=SS*C(IS)
      DO 201 JS=1,3
      ODR(IT,IS,JS)=0.0
      B(IS,JS)=SST*B(IS,JS)*GGI
  201 AS(IS,JS)=HI*A(IS,JS)
      DET=AS(1,1)*AS(3,3)-AS(1,3)*AS(3,1)
      DETI=1.0/DET
      ODR(IT,1,2)= DETI*(AS(3,3)*B(1, )-AS(1,3)*B(3,2))
      ODR(IT,3,2)=-DETI*(AS(3,1)*B(1,2)-AS(1,1)*B(3,2))
      CR(IT,1)=XV(IT,1)+DETI*(AS(3,3)*C(1)-AS(1,3)*C(3))
      CR(IT,2)=XV(1,2)
      CR(IT,3)=XV(IT,3)-DETI*(AS(3,1)*C(1)-AS(1,1)*C(3))
  500 RETURN
  300 WRITE(6,401)
      WRITE(6,402) JN,MT,DELTAT,H( JN,MT),GG(JN,M )
      STOP
  400 FORMAT(10X,5E15.7)
  401 FORMAT(5X, 30H DELTAT,H, OR GG ARE ZERO.                    )
  402 FORMAT(5X,2I5,3E15.7)
  407 FORMAT (5X,3E15.6)
      END
```

```
      SUBROUTINE FS(H)

C     ALL FUNCTIONS COMPUTED HERE ARE A FUNCTION OF H
C     FB(J) ARE THE FUNCTIONS F BAR SUBSCRIP J
C     FBP AND FBPP ARE THE FIRST AND SECOND DERIVATIVES OF FB
C     GPH IS G+H WHERE G=2*H/(H-1)

      COMMON/CF/FB
      COMMON/CFD/FBP,FBPP
      DIMENSION FB(4),FBP(4),FBPP(4),F(4),FP(4),FPP(4)
      HM1= H-1.0

C   THE FOLLOWING IF STATEMENT DETERMINES IF THERE IS A SINGULARITY IN
C   ONE OF THE FB FUNCTIONS.

      IF(HM1) 501,501,101
  101 HM1I=1.0/HM1
      HM1IS=HM1I*HM1I
  100 HI=1.0/H
      HIS=HI*HI
      HS=H*H
      HP1=H+1.0
  102 HP1I=1.0/HP1
      HP1IS=HP1I*HP1I
      HP2=H+2.0
  103 HP2I=1.0/HP2
      HP2IS=HP2I*HP2I
      HP3=H+3.0
  104 HP3I= 1.0/HP3
      HP3IS=HP3I*HP3I
      HP4=H+4.0
  105 HP4I=1.0/HP4
      HP4IS=HP4I*HP4I
      HP5=H+5.0
  106 HP5I=1.0/HP5
      HP5IS=HP5I*HP5I

C     COMPUTE GPH AND ITS DERIVATIVES

      GPH=H*HP1+HM1I
      P=HS-2.0*H-1.0
      GPHP =P*HM1IS

C   COMPUTE F AND ITS FIRST DERIVATIVE FP

```

```
      F(1)=HI-2.0*HP1I+HP2I
      FP(1)=-(HIS-2.0*HP1IS+HP2IS)
      F(2)=HI-4.0*HP1I+HP2I+4.0*HP3I-2.0*HP5I
      FP(2)=-(H          IS-4.0*HP1IS+HP2IS+4.0*HP3IS-2.0*HP5IS)
      F(3)= (HI-4.0*HP1I+6.0*HP        2I-4.0*HP3I+HP4I)
      FP(3)=-(HIS-4.0*HP1IS+6.0*HP2IS-4.0*HP3IS+HP4IS)
      F(4)=2.0*(HP1I-2.0*HP3I+HP5I)
      FP(4)=-2.0*(HP1IS-2.0*HP3IS+HP5IS)
      F(4)=-F(4)
      FP(4)=-FP(4)

C     COMPUTE FB.FBP

      DO 200 I=1,4
      FB(I)=-F(I)*GPH
  200 FBP(I)=-(FP(I)*GPH+F(I)*GPHP)
  500 RETURN

  501 WRITE(6,400) H
      STOP
  400 FORMAT(44H1SINGULARITY IN ONE OF THE F FUNCTIONS
     13H2H=  1F7.4        )
      END
```

```
      SUBROUTINE SOLVE
C     THIS SUBROUTINE SOLVES THE LINEAR EQUATIONS RESULTING FROM THE
C     O BRIEN,HYMAN,AND KAPLAN IMPLICIT FORMULATION. A DIRECT GAUSSIAN
C     REDUCTION IS APPLIED TO THE SPARSE COEFFICIENT MATRIX.
C     THE COEFFICIENT MATRIX HAS THE TRI-DIAGONAL FORM
C     (ODL(I),D(I),ODR(I)) WHERE  I STANDS FOR THE I-TH ROW AND  ODL(I),
C     D(I), AND ODR(I) ARE 3*3 MATRICES.
C     THE SOLUTION IS XV.
      COMMON/INT/NTH,MTH,MS,DELTAT
      COMMON/VAR/XV
      COMMON/CDOD/ODL,D,ODR,CR
      COMMON/TOLERN/TOL
      DIMENSION D(150,3,3),ODL(150,3,3),ODR(150,3,3),CR(150,3)
      DIMENSION XV(150,3)
      DIMENSION A(3,3),AI(3,3),B(3,3),UM(3,3),Y(3)
      WRITE(6,405)((ODR(3,I,J),J=1,3),CR(3,I),XV(3,I),I=1,3)
C
C     GAUSSIAN REDUCTION BEGINS
      DO 205 II=1,NTH
      DO 205 I=1,3
  205 XV(II,I)=CR(II,I)
C     AT THIS POINT XV IS RIGHT SIDE OF EQUATION.
      DO 206 I=1,3
      DO 206 J=1,3
      UM(I,J)=0.0
      IF(I.EQ.J) UM(I,J)=1.0
      D(1,I,J)=UM(I,J)
      ODL(1,I,J)=-ODR(1,I,J)
  206 CONTINUE
      DO 200 II=2,NTH
      MII=II
      DO 201 I=1,3
      DO 207 J=1,3
      D(II,I,J)=UM(I,J)
      ODL(II,I,J)=-ODR(II,I,J)
      DO 204 K=1,3
  204 D(II,I,J)=D(II,I,J)+ODR(II,I,K)*ODR(II-1,K,J)
  207 A(I,J)=D(II,I,J)
      DO 201 K=1,3
      XV(II,I)=XV(II,I)+ODR(II,I,K)*XV(II-1,K)
  201 CONTINUE
C      INVERS COMPUTES THE INVERSE OF THE 3*3 MATRIX A.
      CALL INVERS(A,AI,TOL)
      IF(II.NE.3) GO TO 300
      WRITE(6,404) A
```

```
      WRITE(6,404) AI
      WRITE(6,404) (XV(3,I),I=1,3)
  300 CONTINUE
      DO 202 I=1,3
      DO 203 J=1,3
      A(I,J)=0.0
      B(I,J)=0.0
      Y(I)=0.0
      DO 203 K=1,3
      A(I,J)=A(I,J)+AI(I,K)*D(II,K,J)
  203 B(I,J)=B(I,J)+AI(I,K)*ODR(II,K,J)
      DO 202 K=1,3
  202 Y(I)=Y(I)+AI(I,K)*XV(II,K)
      DO 208 I=1,3
      YAV=ABS(Y(I))
      IF(YAV.LT.1.0E-12) Y(I)=0.0
      XV(II,I)=Y(I)
      DO 208 J=1,3
      AAV=ABS(A(I,J))
      IF(AAV.LT.1.0E-12)A(I,J)=0.0
      BAV=ABS(B(I,J))
      IF(BAV.LT.1.0E-12)B(I,J)=0.0
      D(II,I,J)=A(I,J)
      ODR(II,I,J)=B(I,J)
      ABV=ABS (A(I,J)-UM(I,J))
      IF(ABV.GE.1.0E-12)  GOTO 502
  208 CONTINUE
  200 CONTINUE
C
C     THE GAUSSIAN REDUCTION IS COMPLETE. ANY ELEMENT LESS THAN .0001
C     HAS BEEN REPLACED BY 0.0.
C
C     BEGIN BACK SUBSTITUTION.
C
      DO 220 IB=1,MTH
      L=MTH-IB
      DO 221 I=1,3
      DO 221 K=1,3
  221 XV(L,I)=XV(L,I)-ODR(L,I,K)*XV(L+1,K)
  220 CONTINUE
C
C     SOLUTION COMPLETED.
C     XV IS NOW THE SOLUTION.
C
C     CHECK SOLUTION
```

```
C
      DO 211 II=1,NTH
  211 CONTINUE
      MII=1
      DO 209 I=1,3
      Y(I)=XV(1,I)
      DO 210 K=1,3
  210 Y(I)=Y(I)-ODL(1,I,K)*XV(2,K)
      ABSV=ABS(Y(I)-CR(1,I))
      IF(ABSV.GE.1.0E-05) GOTO 503
  209 CONTINUE
      DO 212 II=2,NTH
      MII=II
      DO 212 I=1,3
      Y(I)=XV(II,I)
      DO 213 K=1,3
  213 Y(I)=Y(I)+ODL(II,I,K)*XV(II-1,K)-ODL(II,I,K)*XV(II+1,K)
      ABSV=ABS  (Y(I)-CR(II,I)        )
      IF(ABSV.GE.1.0E-05) GOTO 503
  212 CONTINUE
      MII=NTH
      DO 214 I=1,3
      Y(I)=XV(NTH,I)
      DO 215 K=1,3
  215 Y(I)=Y(I)+ODL(NTH,I,K)*XV(MTH,K)
      ABSV=ABS(Y(I)-CR(NTH,I))
      IF(ABSV.GE.1.0E-05) GOTO 503
  214 CONTINUE
  500 RETURN
  501 THATA=(MII-1)*DELTAT
      MS=MII
      WRITE(6,400) MS,THATA
      WRITE(6,408)
      WRITE(6,407)((U(MII,I,J),J=1,3),CR(MII,I),I=1,3)
      WRITE(6,406) ((ODL(MII,I,J),J=1,3),(ODR(MII,I,J),J=1,3),I=1,
      STOP
  502 THATA=(MII-1)*DELTAT
      MS=MII
      WRITE(6,401) MS,THATA
      STOP
  503 THATA=(MII-1)*DELTAT
      MS=MII
      WRITE(6,402) MS,THATA
      WRITE(6,404) I,(CR(MII,IC),IC=1,3)
      STOP
```

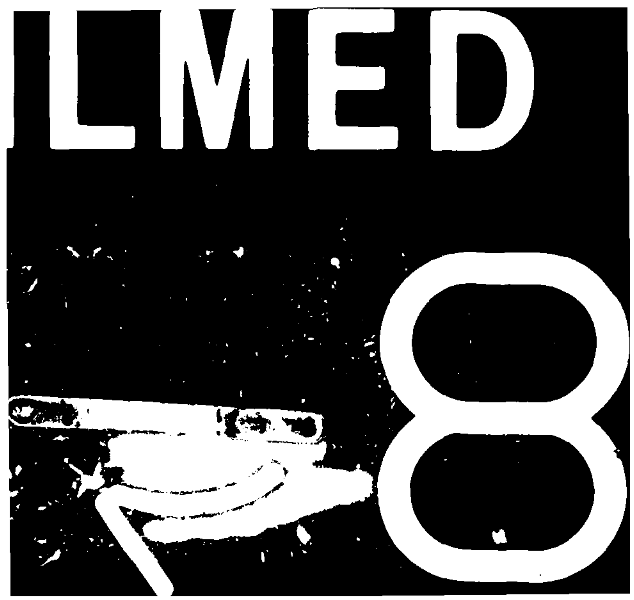
```
C      NUSOL MEANS NO SOLUTION WAS OBTAINED FOR D WAS EITHER NOT THE
C      IDENTITY MATRIX OR  A SUBSTITUTION OF THE SOLUTION DID NOT
C      YIELD THE CORRECT RIGHT SIDE.
   400 FORMAT (33H2 MATRIX IS SINGULAR AT STATION =     1I5,5X,8H THATA =
      $,1F10.4)
   401 FORMAT( 49H2NO SOLUTION FOR D IS NOT IDENTITY AT STATIUON=      ,
      $1I5,5X,7H THATA=   ,1F10.4)
   402 FORMAT ( 56H2NO SOLUTION FOR Y IS NOT EQUAL RIGHT SIDE AT STATION
      $=             ,1I5,5X,7H THATA=     ,1F10.4)
   404 FORMAT(///(3E15.7))
   405 FORMAT(5E15.7)
   406 FORMAT(6E15.7)
   407 FORMAT(5X,4E15.7)
   408 FORMAT(5X,45H FROM SOLVE D, CR, UDL, AND UDR GOOD LOO
       END
```

DTNSRDC ISSUES THREE TYPES OF REPORTS

1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECHNICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.

2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIMINARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE. THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.

3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR INTERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE BASIS.


LMED
8